Titre: Title:	Uncertainty Modeling in Robotic Maintenance Using Computer Vision
Auteur: Author:	Loïc Delgado Sedano
Date:	2019
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Delgado Sedano, L. (2019). Uncertainty Modeling in Robotic Maintenance Using Computer Vision [Master's thesis, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/4128/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/4128/
Directeurs de recherche: Advisors:	Sofiane Achiche, & Maxime Raison
Programme: Program:	Génie mécanique

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Uncertainty Modeling in Robotic Maintenance Using Computer Vision

LOÏC DELGADO SEDANO
Département de génie mécanique

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
Génie mécanique

Décembre 2019

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Ce mémoire intitulé :

Uncertainty Modeling in Robotic Maintenance Using Computer Vision

présenté par **Loïc DELGADO SEDANO**
en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
a été dûment accepté par le jury d'examen constitué de :

David SAUSSIÉ, président
Sofiane ACHICHE, membre et directeur de recherche
Maxime RAISON, membre et codirecteur de recherche
Mélissa CÔTÉ, membre

ACKNOWLEDGEMENTS

I would like to thank my supervisors Prof. Sofiane Achiche and Prof. Maxime Raison, professors at the Department of Mechanical Engineering at Polytechnique Montreal, for their support, their valuable advice and the opportunity they gave me to be a part of such a major project. Both of you are inspiring and lead your students towards success and self-improvement.

I would like to show my gratitude to all the partners involved in this project namely Rolls-Royce Canada Limited, AV&R, the National Research Council and the other students and professors at Université Laval and Polytechnique Montreal. This research would not have been as rewarding without all our fruitful meetings and knowledge sharing. I would like to give a special thanks to AV&R team for all the experience and skills I have gained from them and the many interesting discussions that have helped to advance my research. This work was supported by Mitacs through the Mitacs Accelerate program. Therefore, I am grateful for their financial support and the training workshops they offered me.

A special gratitude for all my colleagues of the Laboratoire de Conception de Systèmes Intelligents et Mécatroniques (CoSIM) for the friendly spirit they create and the inspiration they provide.

Finally, I would like to thank my family and my friends from Brussels and Montreal for the happiness you bring me and your perpetual encouragement. Especially my twin sister, Sarah, one of the most persevering women I know. Keep it up, you will overcome mountains. To Cecilia, I owe you a special thank for the patience you have shown, your continuous support and your love. This adventure would not have been so thrilling without you.

RÉSUMÉ

L'inspection visuelle automatisée est une technologie en expansion dans le domaine industriel, notamment pour l'identification et la réparation de pièces aéronautiques critiques défectueuses. Pour mesurer des défauts de surface, des images prises par un ou plusieurs systèmes d'acquisition sont traitées. Le traitement de ces images requiert de nombreux algorithmes de détection, de recalage, de segmentation et de classification des défauts observés. Dans le cas d'une acquisition robotisée, la cinématique du robot doit également être connue et prise en compte. Tous ces systèmes nécessitent une étape préliminaire d'étalonnage pour pouvoir extraire des informations précises des images capturées.

Chaque étape d'un tel procédé possède plusieurs sources d'incertitude dues à des facteurs internes ou externes. L'évaluation de ces incertitudes permet de quantifier la qualité d'une mesure et d'identifier les parties d'un processus ayant une sensibilité accrue. Devant la multitude de systèmes présents dans une technologie d'inspection, l'objectif de ce projet est d'estimer les incertitudes associées au procédé d'acquisition 2D d'AV&R, i.e. l'étalonnage d'une caméra et des algorithmes de détection de défauts associés. Les pièces d'inspection utilisées dans ce projet sont des aubes de soufflante. Plus particulièrement, seuls les défauts proches du bord d'attaque et du pied de l'aube sont considérés, zone considérée comme critique. Huit acquisitions sont prises pour capturer l'ensemble de la zone critique de l'aube.

Pour évaluer les incertitudes des deux méthodes d'étalonnage de camera d'AV&R, la méthode de Monte Carlo est utilisée. Dû à des contraintes de correspondance de référentiels et aux coûts liés à l'implémentation de cette méthode, l'algorithme d'étalonnage du focus et de la profondeur de champ est adapté à un langage de programmation plus approprié. Une méthode d'estimation de la pose de la caméra pour des points coplanaires, appelée Pose from Orthography and Scaling with Iterations (POSIT), est implémentée pour un setup similaire à celui d'AV&R afin de proposer une alternative à leur seconde méthode d'étalonnage. Parmi toutes les méthodes utilisées dans la routine de détection, un algorithme déterministe de détection de contours appelé *Slope Detection* est très sensible. En particulier, trois paramètres clés jouent un rôle important dans la variation du nombre de défauts détectés et du nombre de fausses détections : la taille du filtre, le seuil d'intensité et la taille minimale de défauts détectés. La corrélation entre les deux sorties de détection et les trois paramètres sensibles est quantifiée pour une acquisition. L'importance de chacun des paramètres est évaluée à l'aide d'une régression par forêt d'arbres décisionnels.

Les incertitudes des algorithmes de détection dépendent des incertitudes d'étalonnage de la

caméra. Étant donné que l'ensemble des incertitudes d'étalonnage n'est pas connu, une série d'images obtenues après étalonnage est utilisée pour évaluer l'incertitude de chaque pixel en prenant l'écart-type de l'intensité comme incertitude standard. Cela permet donc de tenir en compte les variabilités présentes au sein des processus d'étalonnage de la caméra dans l'évaluation des incertitudes de détection. L'incertitude du nombre de défauts détectés et du nombre de fausses détection fournis par *Slope Detection* est évaluée au moyen de simulations de Monte Carlo. La sensibilité des incertitudes de détection à différents finis de surface et à la position des défauts sur l'aube est quantifiée. Enfin, l'impact de l'incertitude de chaque acquisition sur les incertitudes de détection est également quantifié.

Les résultats des simulations de Monte Carlo pour l'étalonnage du focus et de la profondeur de champ montrent que l'incertitude d'intensité ajoutée aux pixels ne peut dépasser 7 sans obtenir des résultats hors des tolérances fixées par AV&R. L'implémentation de POSIT donne la pose de la caméra avec une bonne précision. L'évaluation de l'incertitude de cette pose a permis d'établir que l'incertitude de la longueur focale et des points 3D de la cible d'étalonnage a un impact conséquent sur la coordonnée de profondeur du vecteur de translation. Les deux autres coordonnées de ce vecteur sont surtout sensibles au centre de l'image. L'orientation de la caméra est principalement sensible aux incertitudes de mesure des points 3D et de leurs images. L'incertitude de cette orientation est très élevée à cause du bruit important ajouté aux points 3D. Au vu de la précision du setup d'AV&R, l'algorithme POSIT pourrait potentiellement réduire l'incertitude de la pose de la caméra utilisée.

Le nombre de défauts détectés et de fausses détections fournis par *Slope Detection* ne montrent pas de sensibilité accrue à un des trois paramètres clés. Néanmoins, la taille du filtre et du seuil choisis affectent plus les résultats que la taille minimale des défauts. Toutefois, un jeu de données exempt de valeurs aberrantes permettrait de mieux caractériser cette sensibilité et d'identifier les valeurs optimales des paramètres clés. La méthode de Monte Carlo appliquée à *Slope Detection* montre que les incertitudes de détection sont faibles. En effet, en moyenne pour chaque aube l'incertitude élargie à 95% est unitaire pour le nombre de fausses détections et de 0.2 pour le nombre de défauts détectés. Trois acquisitions adjacentes de la zone critique présentent des incertitudes de détection accrue comparée aux autres acquisitions. Parmi toutes les acquisitions, une seule a un impact plus conséquent sur l'incertitude du nombre de défauts détectés. Le nombre de fausses détections n'est pas plus affecté par une acquisition en particulier. Enfin, les résultats montrent que la réparation de défauts accroît les incertitudes de détection. Une aube réparée présente une incertitude 5 fois plus élevée pour le nombre de défauts détectés et 2.3 fois plus élevée pour le nombre de fausses détections comparée à une aube non réparée. La précision des réparations effectuées par AV&R doit donc être accentuée afin de réduire le nombre d'artefacts présents sur les images.

ABSTRACT

The automated visual inspection is a fast growing technology used in many industrial applications such as for the identification and the repair of damaged critical aeronautical parts. Measuring surface defects consists in taking images with one or several acquisition systems in order to apply digital image processing techniques. These methods use many different algorithms from detection and image registration to segmentation and classification of defects. For a robot-based acquisition, the robot kinematics must be taken into account. The aforementioned systems need a preliminary step of calibration to extract accurate information from the acquired images.

Each step of this technology contains several sources of uncertainty due to internal and external factors. The evaluation of those uncertainties allows to quantify the quality of a measurement and identify the parts of the process having higher sensitivity to errors. Considering this large number of systems present in an inspection technology, the main objective of this project is to the estimate the uncertainties associated to the 2D acquisition process of AV&R, i.e. the camera calibration and the related defect detection algorithms. The inspection parts used in this project are fan blades. In particular, the assessment of detection uncertainties is focused on the defects near the leading edge and the root of the blade, the most critical zone. Eight acquisitions are acquired to cover the entire zone.

To evaluate the uncertainty of both calibration methods developed by AV&R, the Monte Carlo method is used. Due to technical constrains on reference frames correspondence and the costs involved in the implementation of this method, the focus and depth of field calibration algorithm is adapted to a suitable programming language able to run Monte Carlo simulations. A camera pose estimation algorithm, called Pose from Orthography and Scaling with Iterations (POSIT), is implemented on a setup similar to the one used by AV&R in order to propose an alternative to their second calibration method. Among all the methods involved in the detection routine, a deterministic edge detection algorithm called *Slope Detection* is highly sensitive. In particular, three key parameters play an important role in the number of detected defects and the number of false detections: the filter size, the intensity threshold and the minimum size of detected defects. The correlation between both detection outputs and the three sensitive parameters is quantified for one particular acquisition. The importance of each variable is also estimated using a random forest regressor.

The uncertainty of the detection algorithms relies on the uncertainty of the camera calibration methods. Since not all calibration uncertainties are known, a series of acquired

post-calibration images is used to assess the uncertainty of each pixel considering the intensity standard deviation as the standard uncertainty. Therefore, the variability present within the camera calibration methods is taken into account in the evaluation of the detection uncertainties. The uncertainty of the detected defects and the number of false detections provided by *Slope Detection* is evaluated using Monte Carlo simulations. The sensitivity of the detection uncertainties to different surface finishes and to the defects location on the blade is quantified. Finally, the impact of each acquisition uncertainty on the detection uncertainties is also quantified.

The results of Monte Carlo simulations for the focus and depth of field calibration show that the uncertainty of the added noise in pixels cannot exceed an intensity of 7 or risk of obtaining results outside the tolerance zone set by AV&R. The implementation of POSIT provides a camera pose with good accuracy. The uncertainty evaluation of the camera pose allowed to state that the uncertainty of the focal length and the 3D points of the calibration target has a significant impact on the depth coordinate of the translation vector. The other two coordinates are mainly sensitive to the position of the image center. The camera orientation is mostly sensitive to the uncertainty of the measured 3D points and the corresponding image points. In view of the accuracy of AV&R's setup, the POSIT algorithm could potentially reduce the pose uncertainty of the camera used.

The number of detected defects and false detections returned by the algorithm *Slope Detection* do not show a strong sensitivity to any of the three key parameters. Nevertheless, the size of the filter and the threshold have a higher impact on the results than the minimum defect size. However, the generation of a dataset without outliers is required to better characterize this sensitivity but above all to identify optimal values of the key parameters that would give better detection results. The Monte Carlo method applied to the *Slope Detection* algorithm highlights that the detection uncertainties are low. Indeed, for each fan blade the expanded uncertainty at 95%, on average, is unitary for the number of false detections and around 0.2 for the number of detected defects. Three adjacent acquisitions of the blade critical zone present increased detection uncertainties compared to the other acquisitions. Among all the acquisitions, one has significant effect on the uncertainty of the number of detected defects. The number of false detections is not more affected by a particular acquisition though.

Finally, the results show that the defect repair increases the detection uncertainties. A repaired fan blade has a 5 times higher uncertainty for the number of detected defects and 2.3 times higher for the number of false detections compared to an unrepaired blade. The accuracy of the repair process performed by AV&R must therefore be increased in order to reduce the number of artifacts present in the images.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ACRONYMS

2D	Bi-dimensional
3D	Tri-dimensional
CAD	Computer-Aided Design
CCD	Charge Coupled Device
CMM	Coordinate Measuring Machine
CMOS	Complementary Metal Oxide Semiconductor
CNN	Convolutional Neural Network
DLT	Direct Linear Transformation
DOF	Depth of Field
GUM	Guide to the expression of Uncertainty in Measurement
JCGM	Joint Committee for Guides in Metrology
K-NN	K-Nearest Neighbors
MCM	Monte Carlo Method
PDF	Probability Density Function
PnP	Perspective-n-Points
POSIT	Pose from Orthography and Scaling with Iterations
SVD	Singular Value Decomposition
SVM	Support Vector Machine

LIST OF APPENDICES

CHAPTER 1 INTRODUCTION

In the aeronautical field, a major part of incomes of engine manufacturers come from after-sale services [1, 2]. Inspection and repair processes of aeronautical parts are crucial among these services especially for critical elements such as fan blades, shafts, disks or turbine blades. Nowadays, these operations are generally manual hence expensive. To ensure increased quality and safety while reducing costs, many companies head towards partial or complete automation of these processes. However, the standards of inspection and repair of defects were designed for human analysis and interpretation. The robotization of those tasks leads to a modification of these protocols but also to the establishment of human-machine interactions.

These inspection systems are often composed of a robotic arm, carrying an acquisition device on its end effector such as a camera or a laser scanner [3,4]. This combination of such a visual inspection system is called a hand-eye configuration. To extract geometric information from acquired images such as the size of the defects, all these systems need to be calibrated [5–8]. Once this preliminary step is done, several important steps impact the measurement of defects: trajectory planning [9], image registration [10], detection algorithms [11], the projection of defects onto a Computer-Aided Design (CAD) [12] , etc.

Although particular emphasis is given to the accuracy and the repeatability of actual automated processes, a certain uncertainty remains associated with the measurements. It characterizes the dispersion of values of a quantity being measured, the measurand [13]. This uncertainty is quantifiable by means of different techniques described in the literature. Hence, this quantitative criterion of the quality of a measurement is an opportunity for manufacturers developing any accurate measurement technology to highlight it to their customers.

As mentioned, visual inspection routines consist of using a series of different complex systems with numerous uncertainty sources: the environment (temperature, pressure, illumination), the calibration, etc. [13,14] Moreover, in a context of human-machine collaboration, operators working on the parts also represent another source of uncertainty. This latter variability is hard to quantify and is not within the scope of this project [15]. All these uncertainties propagate through the whole process up to the measurement of defects. A slight difference in those measurements can impact significantly the repair zone. Thus, these uncertainties need to be quantified and major sources need to be identified to provide solutions to reduce them if they are out of tolerance. Clearly, the automated removal of defects also involves several sources of uncertainty. Due to the complexity of such a visual maintenance system,

all the uncertainties cannot be evaluated during the short time of a master. Therefore, most relevant steps of the inspection routine need to be studied first. Based on the experience of engineers of both organization partners AV&R and Rolls-Royce Canada Limited, a diagram of the whole inspection process was drawn. This diagram aimed at classifying sources of uncertainty by order of importance. The result showed that 2D and 3D acquisition systems and calibration methods associated seem to contain major sources of uncertainty. This project is thus limited to the study and the quantification of the sources of uncertainty of the 2D acquisition of a fan blade. Precisely, it focuses on the calibration of the 2D device and the related detection algorithms. Two calibration methods developed by AV&R are considered: the calibration of the focus and the depth of field and the external parameters calibration. A pose estimation technique, called POSIT, is developed to overcome the implementation constraints of the latter calibration technique. The detection algorithm mainly taken into account in this work, is called *Slope Detection*. The sensitivity of its outputs to a variation of three key parameters is quantified (namely a filter size, a threshold and the minimum defect size). The uncertainty of both calibration methods and the sensitive detection algorithm is evaluated.

This master's thesis is divided into several sections. Chapter 2 describes the state of the art of calibration methods and detection algorithms but also the study of uncertainty propagation when using those techniques. Afterward, the problematic, the objectives, the hypothesis and the assumptions are stated in chapter 3. Chapter 4 is dedicated to the methods used and the mathematical background needed to understand them. Results obtained from those methods are depicted in chapter 5 and discussed in chapter 6. The limitations and perspectives of this work are also described in the latter. The final chapter summarizes the main results and concludes this thesis.

CHAPTER 2 LITERATURE REVIEW

An accurate defect measurement system requires a well-calibrated detection system with well understood uncertainties. In this literature review, several camera-based calibration methods are introduced. Although there exist several high precision methods, the uncertainty of the results can be estimated. Camera parameters uncertainty propagates to the detection algorithms which, themselves, involve different uncertainty sources. A few detection techniques are thus reviewed. The end goal is to be able to evaluate the order of magnitude of all these uncertainties in order to develop solutions to mitigate them. Different uncertainty evaluation types exist and are thus explained in this following review of the literature.

2.1 2D Acquisition

The inspection system studied in this project is an articulated robot equipped with a camera as illustrated in figure 2.1. There are two ways to calibrate a camera-aided robot as defined in [16]. The first method consists in calibrating the robot and the camera parameters jointly. In the second approach, both the camera and robot calibrations can be dissociated if the camera is already calibrated. The robot pose is then measured several times with the vision system attached to the robot to estimate its parameters.

Figure 2.1 Automated visual inspection system from © Universal Robots 2019 [17]

Due to the large number of systems present in an automated visual inspection technology, the analysis and the uncertainty evaluation of all of them cannot be carried out due to the time and means constraints specific to the realization of a master's project. Hence, this thesis is focused on one of the most impacting steps according to AV&R: the 2D acquisition with a CCD or CMOS camera. This acquisition includes the camera calibration and the implemented defect detection algorithms. The second calibration approach of a camera-aided robot is therefore used and the robot calibration is not taken into account in this work [16]. Furthermore, the uncertainty assessment of the robot has already been done by AV&R. Some of the calibration methods reviewed can also be applied to stereo vision but this state of the art mainly focuses on mono vision systems since it is the kind of device used. The same review structure could be done for the 3D acquisition process in a future work, once it is operational. Indeed, this acquisition device aims at detecting defects in 3D and thus needs to be calibrated. Although the uncertainty evaluation of the 2D acquisition gives information on the defects measurements quality, the 3D part has a lot of uncertainty sources impacting the measurements and needs to be estimated.

2.1.1 Calibration methods

The goal of camera calibration is to determine the relationship between the 3D points of an observed scene and the 2D coordinates of their image. This relationship is needed when metric information from images is required such as the size of an object. A common and simple modeling is the pinhole for which the lens is not considered in the camera (perspective projection) [18]. The assumption made is that the CCD sensor of the camera is a 2D plane on which the image is projected. To project a point $\mathbf{M} = [X, Y, Z, 1]^T$ onto the image plane, a homogeneous transformation is applied to determine the corresponding coordinates of the 2D point $\mathbf{m} = [u, v, 1]^T$ (equation 2.1). This equation is valid for a scale factor λ [18]. The pinhole model is illustrated in figure 2.2.

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K} \begin{bmatrix} \mathbf{R} & | & \mathbf{t} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.1}$$

The matrix $\mathbf{K}$ contains intrinsic parameters of the camera, i.e. its optical configuration (equation 2.2). Those different internal parameters are described in table 2.1. To express the focal distance in pixels, scale factors c_u and c_v are used. They are equal to 1 if pixels are squared [19].

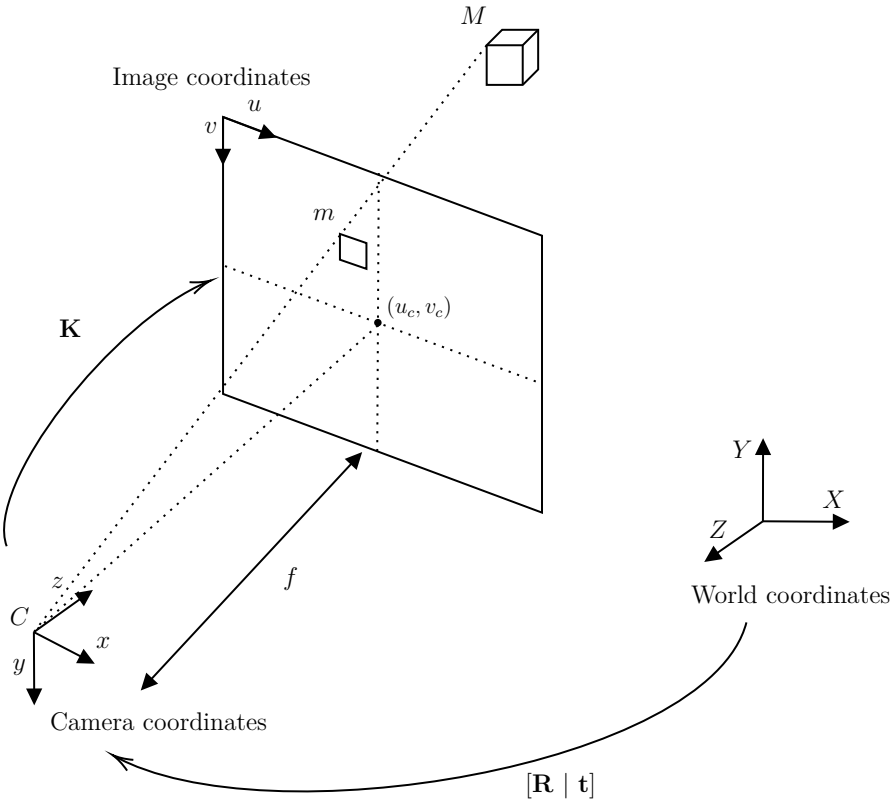


Figure 2.2 Schematic representation of the pinhole model reproduced with permission from [18] © 2015 IEEE

$$\mathbf{K} = \begin{pmatrix} c_u f & \alpha & u_c \\ 0 & c_v f & v_c \\ 0 & 0 & 1 \end{pmatrix} \tag{2.2}$$

External or extrinsic parameters refer to the position and orientation of the camera (pose) with respect to a certain reference frame called *World*. To express a point in the camera reference frame, a rotation and a translation need to be applied. These transformations are gathered in a 4x4 homogeneous transformation matrix $\mathbf{P}$. This matrix is composed of a 3x3 rotation matrix $\mathbf{R}$ and a 3x1 translation vector $\mathbf{t} = [t_x, t_y, t_z]^T$(equation 2.3).

$$\mathbf{P} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{pmatrix} \tag{2.3}$$

A change of the camera pose results in a modification of the extrinsic parameters while intrinsic parameters remain constant because the optical configuration does not change [18].

Table 2.1 Description and symbol of each intrinsic parameter [19]

Parameter	Symbol	Description
Focal distance	f	Distance between the image plane and the optical center of the camera, i.e. the origin of the camera reference frame
Scale factor (x-axis)	c_u	Linear pixel density along the x-axis of the CCD sensor
Scale factor (y-axis)	c_v	Linear pixel density along the y-axis of the CCD sensor
Principal point	(u_c, v_c)	Intersection between the line along the focal distance and the image plane
Skew coefficient	α	Non-orthogonality of the CCD sensor axes

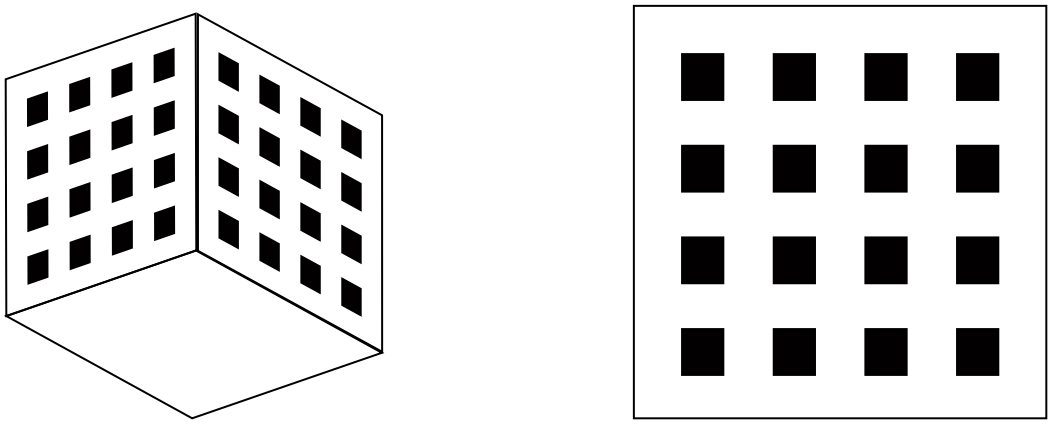
Thus, there are 6 intrinsic parameters and 6 extrinsic parameters to be determined to solve the calibration problem. Indeed, the rotation matrix can be represented by 3 Euler angles [20]

Several classifications exist among calibration methods as mentioned by Joaquim Salvi et al. [21]. Calibration techniques can be implicit or explicit depending on the physical meaning of the calibrated parameters. If they correspond to internal and external parameters aforementioned, it is an explicit calibration. This physical meaning is not required if one is only interested in 3D reconstruction for example [22]. Explicit and implicit methods can be linear or nonlinear depending on imperfections added to the model (lens distortion, etc.). Finally, the type of calibration object is also a classification criterion since it can be 3D or planar (figure 2.3). The advantage of 3D calibration objects is the higher accuracy associated with it but at a higher cost.

One of the well-known implicit and linear methods is the *Direct Linear Transformation* or DLT [24]. This method was initially used for the aerial photogrammetric domain. Based on known points in the 3D space, one can determine all 12 parameters describing the relationship between any 3D point and its image in pixels (equation 2.4). These calibration points are often called control points.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} L_1 & L_2 & L_3 & L_4 \\ L_5 & L_6 & L_7 & L_8 \\ L_9 & L_{10} & L_{11} & L_{12} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.4}$$

To normalize the DLT matrix, L_{12} is considered to be unitary. Basically, it is equivalent to



(a) (b)

Figure 2.3 a) 3D calibration object b) Planar calibration target reproduced with permission from [23] © 1987 IEEE

dividing each L_i by L_{12}. Therefore, all these coefficients are defined up to a scale factor. Since the DLT matrix contains 11 unknowns and that each 2D point of coordinates (u, v) provides 2 equations, 6 control points are required to obtain a complete solution (equation 2.5).

$$\begin{cases} u = \frac{L_1X+L_2Y+L_3Z+L_4}{L_9X+L_{10}Y+L_{11}Z+1} \\ v = \frac{L_5X+L_6Y+L_7Z+L_8}{L_9X+L_{10}Y+L_{11}Z+1} \end{cases} \tag{2.5}$$

To isolate the 11 L_i parameters, the equation 2.5 can be rearranged to obtain equation 2.6. Those 11 DLT parameters are computed using singular value decomposition (SVD).

$$\begin{bmatrix} X_i & Y_i & Z_i & 1 & 0 & 0 & 0 & 0 & -u_iX_i & -u_iY_i & -u_iZ_i \\ 0 & 0 & 0 & 0 & X_i & Y_i & Z_i & 1 & -v_iX_i & -v_iY_i & -v_iZ_i \end{bmatrix} \begin{bmatrix} L_1 \\ L_2 \\ \vdots \\ L_{11} \end{bmatrix} = \begin{bmatrix} u_i \\ v_i \end{bmatrix} \tag{2.6}$$

Once all the DLT parameters are obtained, the relationship between them and the extrinsic and intrinsic parameters can be identified [25]. Although the DLT method is frequently used in the literature, its accuracy has been considered non-optimal over the years. It is mainly due to its modeling simplicity. Nevertheless, it allows to provide a good estimate of camera parameters. The main limitation of this technique is that only non-coplanar control

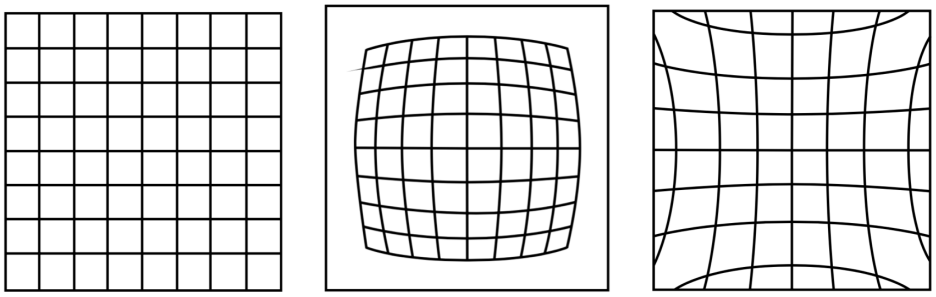
points can be used. If this assumption is violated, the solution of the system is a degenerate case [26]. Many authors based their work on this method in order to develop similar implicit and linear techniques [26, 27]. Those generally require at least 6 non-coplanar points and they use least square regression techniques.

To overcome the shortcomings of linear calibration methods, more accurate nonlinear techniques were proposed [22, 23, 28, 29]. Lens imperfections tend to deform the image. To take them into account, one or several distortion terms need to be added to the classical pinhole model because light rays do not travel along a straight line when they pass through the lens [22]. The comparison of two types of distortion in visual systems is depicted in figure 2.4. These nonlinear models require an initial solution from which an optimization step is run using nonlinear estimation algorithms [30–32]. In spite of their good accuracy, these algorithms demand high computational resources due to the large number of iterations needed to reach convergence. The choice of the initial solution has also an impact for avoiding the algorithm to converge to a local minimum.

To reduce the number of iterations and obtain better computation performance, Roger Tsai (1987) is one of the first authors to introduce a two-step method [23]. He added a coefficient of radial distortion κ_1 to the pinhole model. This distortion is mainly due to the imperfections of the lens curvature. To converge to a global minimum, a part of the parameters is evaluated using a linear method: the rotation matrix $\mathbf{R}$, the coordinates t_x, t_y and both scale factors c_u and c_v [33]. The focal distance f and the coordinate t_z are then estimated considering κ_1 equal to 0 first. A nonlinear minimization is then carried out to compute the exact value of those three parameters. Several methods exist and aim at minimizing the distance between image points and the projection of 3D points with the model implemented [21]. A common optimization algorithm is named Levenberg-Marquardt [33]. Limitations of Tsai's algorithm lie in the requirement of having a sufficient number of non-coplanar control points (>7). It can be a disadvantage when the manufactured calibration target has few control points.

If the non-coplanarity is not fulfilled, one can only compute a submatrix of the calibration matrix [26]. The equation 2.7 shows the relationship between the distorted point $P_d = (X_d, Y_d)$ and the corrected one $P_u = (X_u, Y_u)$. The terms δ_x and δ_y represent the added distortion. This radial distortion on the image plane is illustrated in figure 2.5. The radial distance r from the principal point of the image plane is defined in equation 2.7.

$$\begin{cases} X_d + \delta_x = X_u = X_d(1 + \kappa_1 r^2) \\ Y_d + \delta_y = Y_u = Y_d(1 + \kappa_1 r^2) \\ r = \sqrt{X_d^2 + Y_d^2} \end{cases} \tag{2.7}$$



(a) (b) (c)

Figure 2.4 Image distortion examples : a) non-distorted image b) negative distortion (*barrel*) c) positive distortion (*pincushion*) reproduced with permission from [34]

Based on Tsai's work, numerous authors developed similar methods using DLT as a preliminary step [35, 36]. Janne Heikkilä (2000) noted that the non-colinearity of lens curvature centers required to add another term of decentering distortion also called tangential distortion [37]. A radial distortion term of order 4 and two tangential distortion coefficients p_1 and p_2 are added to the equation 2.7 to obtain a more complete model with distortion (equation 2.8)

$$\begin{cases} X_d + \delta_x = X_u = X_d(1 + \kappa_1 r^2 + \kappa_2 r^4) + D1 \\ Y_d + \delta_y = Y_u = Y_d(1 + \kappa_1 r^2 + \kappa_2 r^4) + D2 \\ r = \sqrt{X_d^2 + Y_d^2} \end{cases} \tag{2.8}$$

$$\begin{bmatrix} D1 \\ D2 \end{bmatrix} = \begin{bmatrix} 2p_1 X_d Y_d + 2p_2(r^2 + 2X_d^2) \\ p_1(r^2 + 2Y_d^2) + 2p_2 X_d Y_d \end{bmatrix} \tag{2.9}$$

Figure 2.6 illustrates the effect of tangential distortion when observing projected lines on the image plane. The specific feature of Heikkilä's method is the use of circular control points. They are generally perceived as ellipses on the image plane when the perspective projection model is used [37]. The mathematical development is based on this particular shape.

To refine the previous model, a third distortion term can be added: the *thin prism* parameter. It describes manufacturing defects and camera mounting imperfections. This parameter is composed of radial and tangential distortions [36]. The main difference compared to the two previous methods is that the nonlinear minimization algorithm is run with the whole set of parameters.

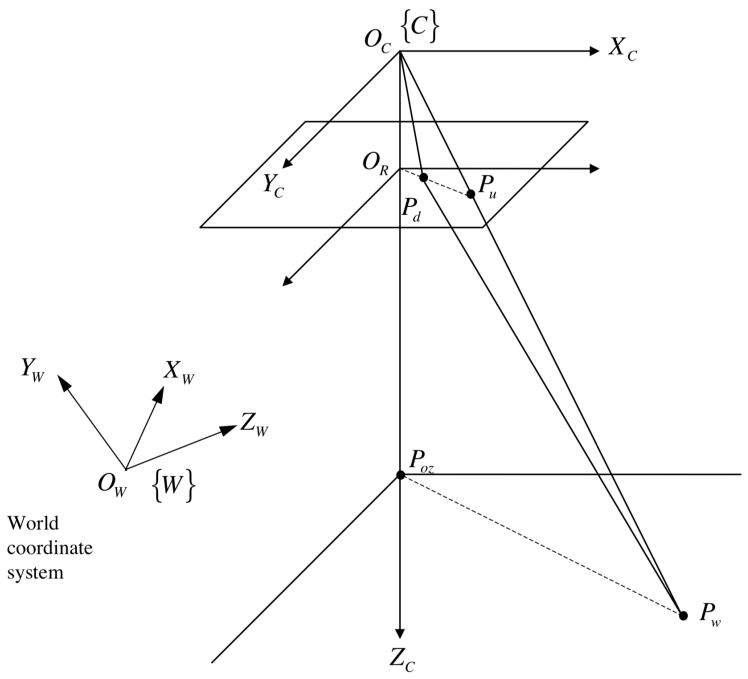


Figure 2.5 Effect of radial distortion : the point P_d is not projected in a straight direction reproduced with permission from [21] © 2002 Elsevier

One of the recent and most implemented methods is introduced by Zhang, taking only radial distortion into account [38]. Unlike previous authors assuming that the axes of the CCD sensor are perpendicular ($\alpha = 0$), Zhang considers it as incorrect. He added thus a non-zero skew coefficient which is a linear distortion term. To simplify computations, Zhang assumes that the calibration target is located at Z = 0. Afterward, a nonlinear minimization based on the maximum likelihood criterion is suggested.

Recently, several authors described techniques using a few number of control points. To solve the equation system 2.1 and obtain a non-degenerate solution, some assumptions need to be made to satisfy the required number of constraints. A reduced number of control points can be used if only intrinsic parameters are computed [39] or if one only seeks to estimate the pose of the camera [40,41]. The estimation of camera external parameters, frequently studied in the literature, is called "Perspective-n-points" (PnP) problem, where n is the number of control points [42]. If the entire set of parameters needs to be evaluated, some of them need to be known beforehand. One can use a single image of 4 coplanar points and 1 non-coplanar to calibrate the camera [43]. To this end, one assumes that the principal point is at the image center, the sensor axes are perpendicular and that $c_u f = c_v f$. In spite of the implementation of a linear step followed by a nonlinear minimization, this method is less robust to noise

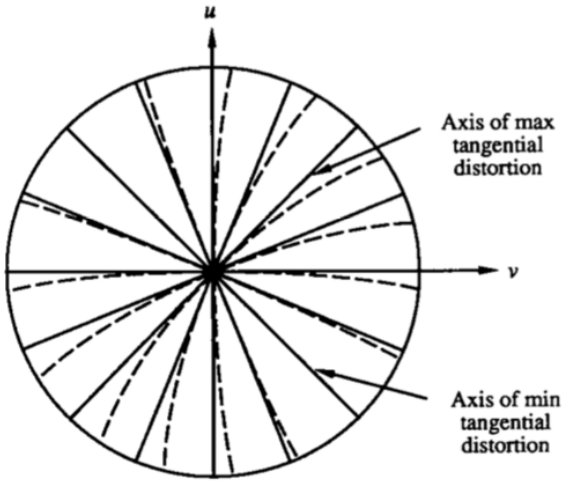


Figure 2.6 Impact of the tangential distortion on projected lines reproduced with permission from [36] © 1992 IEEE

than Tsai's or Heikkilä's for example. Moreover, its accuracy is lower because there is no distortion term added to the model.

Some authors proposed to use only coplanar points to compute intrinsic and extrinsic parameters of the camera [44]. This type of technique allows to opt for planar calibration targets such as chessboards rather than an expensive 3D object (figure 2.3). Chatterjee et al. do not consider the Z coordinate of the 3D point by placing the origin of *World* reference frame onto the calibration target. The three types of distortion are introduced in the model. These methods also evaluate a part of the parameters with a linear technique while the rest is determined with an iterative algorithm. The greatest limitation is the presence of degenerate configurations. Some of them do not provide a relevant solution for the focal distance and t_z. One example is when the optical axis is perpendicular to the calibration object. In that case, only the ratio of those latter parameters can be computed.

It exists a category of calibration methods that do not require the use of a calibration object with known 3D control points. It is frequently named self-calibration or auto-calibration. This technique involves determining intrinsic parameters of the camera based on correspondence points in multiple views [45]. The working principle of self-calibration rests upon an absolute conic Ω_∞ and on existing constraints of all parameters. This conic is defined on an infinite plane and is invariant to a rigid transformation, i.e. a change of the camera pose [46, 47]. The projection of this absolute conic onto the image plane is also invariant and directly related to intrinsic parameters [48]. Figure 2.7 depicts this conic and its image ω. Initially, auto-calibration methods were based on Kruppa equations developed by Faugeras

et al. [47]. Others suggested to build an absolute quadric which is a degenerate quadric since it is any plane tangent to the absolute conic [49]. Despite the simplicity of self-calibration, it is less accurate than using a calibration object. Once the intrinsic parameters calibrated, the pose of the camera can be evaluated using the PnP problem principle [42].

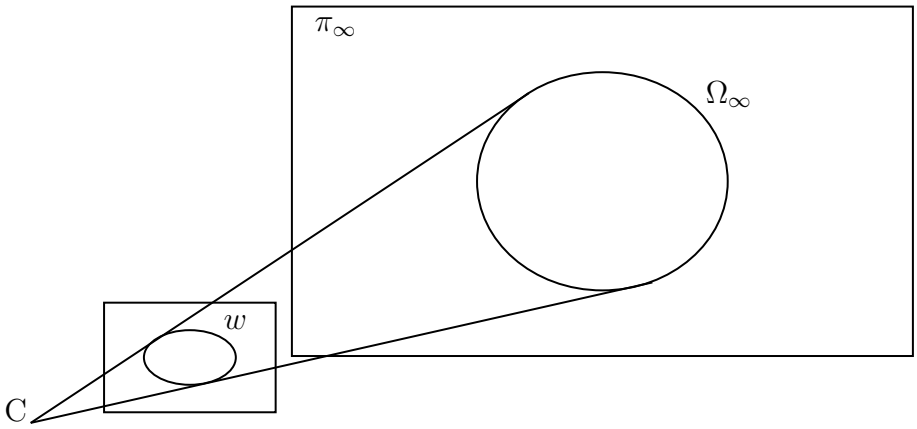


Figure 2.7 Absolute conic and its image reproduced with permission from [46] © 2003 IEEE

Previous methods rest upon constraints of the observed scene but also intrinsic parameters constraints. It is possible to use the motion of the camera to extract new constraints [46, 50]. In fact, intrinsic parameters can be evaluated using a pure rotation motion [51, 52] or a pure translation [53].

Rarely, some papers in the literature refer to 1D calibration methods [54, 55]. They use for instance, an object composed of three collinear points in rotation around one of them [55] or two orthogonal 1D objects [54]. In the first case, the *World* reference frame and the camera reference frame are the same, i.e. the matrix of external parameters $\mathbf{P}$ is the identity matrix. Afterward, the same aforementioned calibration algorithms are used. The drawback of this technique is the requirement of at least 6 points of view or observations as denoted in [55]. Moreover, some motions of the 1D object along quadratic curves are degenerate cases. A degenerate case occurs when one of the 6 points of view is dependent on the 5 other observations. It creates a singularity that leads to false calibration results. All parameters cannot be computed with the second method and their accuracy is poor.

2.1.2 Automated defect detection

Although the literature is full of detection algorithms, this state of the art focuses on techniques aiming at detecting surface defects such as those observed on aeronautical parts as it is the scope of this project. Furthermore, numerous non-destructive inspection techniques

exist but those studied in this project use at least a CCD or CMOS camera [56]. When an anomaly is detected on a part, it needs to be classified because repair processes depend on the type of defect. There are lots of different defects that differ in size, in depth, etc. The main studied defects are nicks, dents, erosion, scoring, galling, tip curls, etc. [57]. Figure 2.8 illustrates the existing variability among those defects. The goal of defect detection is to identify which of them are acceptable, i.e. in the tolerance zone and those that need to be repaired.

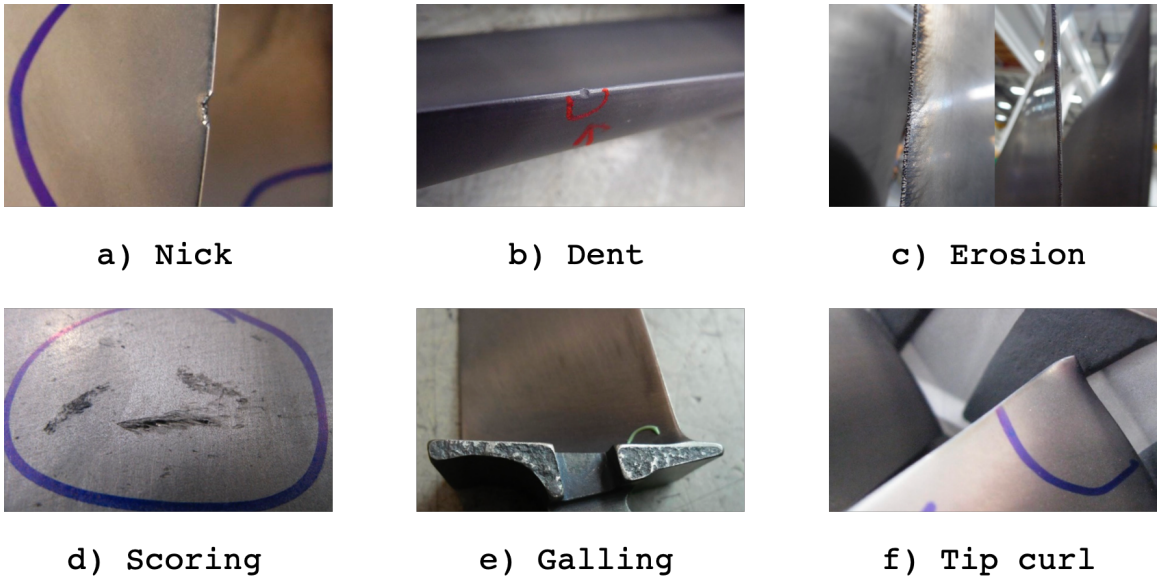


Figure 2.8 Usual defects observed on an aeronautical part reproduced with permission from [57]

There are two main categories of detection techniques : classical methods using image processing and those based on machine learning. Classical methods of detection can be classified in statistical, structural and filter-based approaches [58]. Statistical methods rely on the study of pixel distribution [59]. Histogram equalization and threshold techniques are examples [60]. Structural methods consist of edge detection algorithms [61] and morphological operations [62]. Filter-based methods can be applied in the spatial domain or in the frequency domain. The term filter is used in the broad sense because it includes different operators such as Sobel, Laplacian and Gaussian [63]. In the frequency domain, common known methods such as Fourier transform or wavelet decomposition can be used [64]. All these techniques have the specific feature of not requiring any prior knowledge of the defects unlike the techniques based on pattern matching. The latter consist in identifying all defects and comparing them with a database to evaluate the correlation [59, 65]. Generally, a combination of the previous techniques are used to create a complete defect detection algorithm.

Once all defects are identified and segmented, they need to be classified. This classification requires a particular expertise on the defects and can be done by hand or by means of patterns characterizing those defects [63].

Recently, many authors noted that image processing techniques are more sensitive to false detections [66]. To increase the accuracy of defect detection, they used the photometry; a principle based on light reflection which consists in illuminating an object from multiple directions to reconstruct it in 3D using integration methods [66, 67]

With the rise of machine learning in computer vision [68–70], several authors developed their own solution in the defect inspection field since it allows to detect more complex defects. Conventional machine learning algorithms are divided into two categories: feature extraction and defect classification [71]. Any relevant information on the defects such as the edges, for example, is considered to be a feature and is gathered in a feature vector. As mentioned in [72], feature extraction can be done using the conventional image processing techniques. The classification of defects consists in assigning the correct label to a defect. In this work, the classes could be defined according to the observed defects as seen in figure 2.8. Among all the classification methods existing in the literature, Support Vector Machine, K-Nearest Neighbors and fuzzy logic are frequently used [72].

Due to the sensitivity of usual image processing techniques to light changes and noise [71] and to improve the performance of conventional machine learning algorithms methods based on convolutional neural networks (CNN) were developed [73]. The power of deep neural networks lies in the combination of feature extraction and classification in one network [74]. Such a neural network aims at reproducing the neuronal connections of our brain by stacking different layers made of several nodes [74, 75]. The nodes of a layer n are obtained by combining the nodes of the previous layer $n-1$ through a weighted sum. If one wants to classify defects on images, the input layer is thus the image while the outputs are the different classified defects [76]. In between, there are hidden layers. Yann LeCun et al. (2015) defines them as units non-linearly deforming the input to produce linear separable outputs for the classification [76].

To ensure a correct classification of detected defects, the neural network needs to be trained which corresponds to finding the optimal weights of each layer [74]. Therefore, one needs to feed this network with a set of images containing defects, i.e. the training set. The dataset is divided into two subsets namely the training set and the validation step to avoid overfitting. Overfitting happens when the network neurons co-adapt resulting in a high interdependency [77]. Hence, the model fits to the noise rather than on data only. The validation set is used during the validation step which aims at characterizing the quality of

the implemented model. Two training approaches exist for defect detection. We can detect anomalies in which case the training database is composed of non-damaged samples [78] or use only defects that we want to classify as training set [79]. In the first approach, any image different from a non-damaged sample will be considered as defective. It allows to detect any type of defects even the non-classified ones. Nevertheless, this method is more prone to false positives, i.e. classification of artifacts as defects. The second approach is more suitable when there is a different repair protocol for each defect, as it is the case in this project. Indeed, if the first approach is used one will end up with an unknown type of defect for which a repair procedure is not defined. Furthermore, the second approach requires a large database of all possible defects in order to correctly train the network and avoid the overfitting. In some cases, it is possible to use data augmentation though [73]. Data augmentation consists in applying invariant transformations to the actual dataset to create images that will be considered completely new for the neural network. There are network architectures like cascade auto-encoders designed for detection but that can be used for fast classification of defects when a compact CNN is added to it [71].

In this project, only classical image processing techniques were initially implemented on AV&R's detection software. Machine learning is thus not used yet but it is in the process to be implemented. It requires a more detailed literature review and a comparative study between machine learning and image processing techniques to conclude on the potential improvement of detection results. This whole work could be the center of a master project.

2.2 Uncertainty evaluation

The Joint Committee for Guides in Metrology (JCGM) defines the concept of uncertainty in the Guide to the expression of Uncertainty in Measurement (GUM). It is a non-negative term characterizing the dispersion of values of a physical quantity being measured, i.e. a measurand. This dispersion occurs because the result of a measurement is an estimate of the value of this physical quantity. In other words, the (measurement) uncertainty represents a doubt existing on the value of the measurement even after some correction factors have been applied. To take this lack of knowledge into account, the estimate of the measurand is associated to an uncertainty. To define the standard uncertainty associated to the output estimate of a model, one needs to set the relationship between N input parameters X_i (i = 1,.. N) and the measurand Y (equation 2.10) [13].

$$Y = f(X_1, X_2, ...X_N) \tag{2.10}$$

In fact, the input parameters and the measurand cannot be directly measured, only an estimate of those can be evaluated : y and x_i (equation 2.11).

$$y = f(x_1, x_2, ...x_N) \tag{2.11}$$

There are two ways to evaluate a standard uncertainty commonly known as type A uncertainty and type B uncertainty [13]. Type A uncertainty is based on a statistical analysis of several independent observations of the same process and defines the arithmetic mean $\bar{y}$ as the best estimate of the expectation of a random variable (equation 2.12). Differences in observed values y_i are due to random effects or non corrected systematic errors [13].

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{2.12}$$

$$s(\bar{y}) = \frac{\sqrt{\sigma^2_{exp}}}{n} \tag{2.13}$$

$$\sigma^2_{exp} = \frac{1}{n-1}\sum_{j=1}^{n}(y_j - \bar{y})^2 \tag{2.14}$$

The standard uncertainty $u(y_i)$ is thus defined as the standard deviation of the arithmetic mean $s(\bar{y})$ (equation 2.13) where σ^2_{exp} is the experimental variance of the probability distribution of y defined in equation 2.14.

The same reasoning can be conducted for input parameters for which there is also a standard uncertainty. The JCGM notes that a large number of independent observations are required to guarantee consistent estimates for the arithmetic mean and the standard deviation. Type B uncertainty refers to any uncertainty not evaluated as a type A uncertainty and based on known information on the variability of the studied random variable. The evaluation of the standard uncertainty can rely on calibration certificates, specifications of the manufacturer or any other knowledge of the uncertainties or the distributions associated to variables of the process. In an industrial project like this one, the uncertainty evaluation type chosen depends on what information is available on the studied models. If there is no information for a particular system, the assessment of uncertainty is carried out by analyzing a series of independent observations, i.e. as a type A. In this project, type B evaluation was mainly used since distributions or input uncertainties of the 2D acquisition processes could be extracted. Nevertheless, type A uncertainty evaluation was used for the calibrated images.

The combined standard uncertainty $u_c(y)$ of the estimate of a measurand y is obtained by

combining the standard uncertainty of estimates of input parameters x_i. One considers them independent otherwise a term of covariance needs to be added in the computation of the combined standard uncertainty. The combined standard uncertainty of an estimate y, defined in the equation 2.15, is a weighted sum of the standard uncertainty of estimates x_i, noted $u(x_i)$. Those uncertainties can be evaluated as type A or type B uncertainties. Weighted coefficients are called sensitivity coefficients.

$$u_c(y) = \sqrt{\sum_{i=1}^{n} \left(\frac{\partial f}{\partial x_i}\right)^2 u^2(x_i)} \tag{2.15}$$

It represents a first-order Taylor series expansion because the function f is considered to be nonlinear. Higher order terms can be added to model the nonlinearity of the relationship between Y and X_i if needed [13].

To expose clearly the measurement uncertainty, one can define an interval in which lies the expected set of measurement results. Therefore, the expanded uncertainty U is defined using a coverage factor k. The result of a measurand Y is then included in this interval with a certain level of confidence as shown in equation 2.16.

$$\begin{cases} U = k u_c(y) \\ y - U \leq Y \leq y + U \end{cases} \tag{2.16}$$

The level of confidence of this interval is directly related to the coverage factor k. This factor is a parameter higher than 1 and chosen regarding the desired level of confidence. For instance, a coverage factor of 2 guarantees that the result of a measurement encompasses 95% of the probability distribution of the output quantity. There are three manners to assess the uncertainty of the estimate of a measurand [80]:

Analytical methods consist in obtaining the probability density function of Y through mathematical developments

The law of uncertainty propagation stated in the GUM and based on Taylor series approximation (with or without higher order terms)

Numerical methods based on the propagation of distributions

In the Supplement to the GUM, the JCGM suggests an alternative option to the uncertainty assessment method described in the GUM: Monte Carlo method (MCM) [80]. This approach is based on the propagation of probability distributions rather than uncertainty propagation. It should be noted that the mathematical development of this method considers only a

model with multiple inputs and a unique output. Monte Carlo is generally used when the method described in the GUM provides inconsistent results due to the nonlinearity of the studied model or if the output parameter does not follow a normal distribution or a "scaled and shifted t-distribution" [80]. This technique consists in associating a probability density function (PDF) to each input parameter. These PDF will be propagated through the model as depicted in figure 2.9. Practically, for each parameter X_i a probability distribution is associated to it if it is independent and a joint distribution if it is not the case. We can choose the number N of trials and for each trial, a value of the PDF of the X_i is sampled. The model is run with those values to provide a certain value of the output. By gathering all these output values, one can form a probability density function for Y. Based on this PDF, an estimate, a standard deviation and an expanded interval can be determined and thus compared with the results obtained with the law of uncertainty propagation.

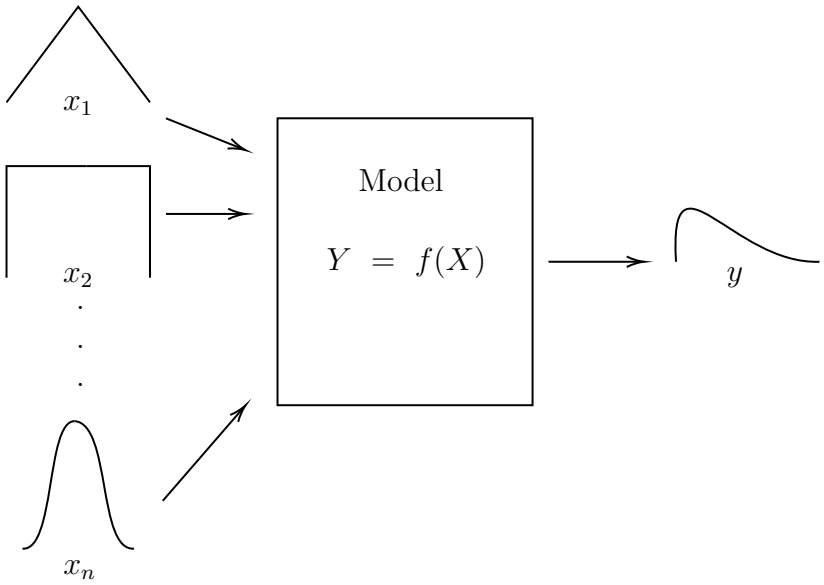


Figure 2.9 Propagation of distributions (Monte Carlo method) reproduced with permission from [80]

The Monte Carlo method is efficient for a high number of trials [81]. The Supplement to the GUM suggests to choose a number N times 10,000 higher than $\frac{1}{1-p}$, where p is the coverage probability or level of confidence as defined in the GUM [13].

2.2.1 Calibration methods uncertainty

In section 2.1.1, many usual calibration methods are described. Some of them are considered as gold standards. The establishment of a standardized and universal approach for assessing

the uncertainty by the JCGM allowed several authors to study the associated uncertainties of these calibration techniques.

Zhu et al. (2009) [82] chose to study the model suggested by Heikkilä and J.Silven (1997) [35], i.e. a nonlinear model with tangential and radial distortions. They noted that the electronic noise of the camera and illumination changes are uncertainty factors that propagate through the perspective projection model giving rise to an uncertainty for each camera parameter. To assess these uncertainties the authors compared an analytical method with Monte Carlo method. Since the control points are the input parameters, their uncertainties need to be estimated first. The assumption made is that one can associate to each control point a Gaussian distribution with a mean value of zero and a standard deviation up to one pixel. This order of magnitude is frequently used as input uncertainty for the control points in the literature [83–85]. The authors are usually interested in the subpixel noise added to the control points. The most important conclusion drawn is that both uncertainty assessment techniques give the same results up to a certain Gaussian noise added.

Other authors carried out similar studies with different calibration models such as DLT [83] or another nonlinear one [86]. In [86], the uncertainty assessment aims at describing the effect of distortion on those uncertainties. They concluded that when distortion is added to the model while the input uncertainties being higher than those distortion coefficients, results were biased. Therefore, it is better to use a model without distortion in that case. However, when distortion is high, the model with distortion provides greater results. Viala et al. (2005) described the assessment of uncertainty when a 3D object is used. The calibration method is not specified but it is a two-step technique consisting in a linear estimation followed by a nonlinear refinement [84]. Noise is added to 3D control points and their images as input uncertainties. The impact of the number of control points on the camera parameters uncertainty is also studied.

H. García-Alfonso and D. Córdova-Esparza (2018) preferred to compare numerical methods of uncertainty assessment on an unspecified calibration model: Monte Carlo method and Latin hypercube sampling [87]. Latin hypercube sampling differs from Monte Carlo in the way input parameters are sampled. Rather than using N random samples of each input probability distribution, this one is divided in N equiprobable intervals. Then, for each interval a sample is taken to be used as a model input value [88]. The goal of this technique is to reduce the number of required iterations to guarantee convergence.

Analytical approaches based on covariance propagation exist and are compared to numerical methods such as the Monte Carlo method [89]. Different uncertainty factors are studied like the illumination, edge effects and scene irregularities.

2.2.2 Detection methods uncertainty

To our knowledge, there is no study of the uncertainties associated to a whole detection process of surface defects in the literature. Nevertheless, some papers introduce the evaluation of the aforementioned algorithms used for detection. The list presented in this section is by no means exhaustive and aims at describing the state of the art about uncertainty in some image processing techniques such as edge detection algorithms and filter-based methods.

H.Zhang et al. (2017) introduced the uncertainty evaluation of a template matching method [90]. They compared the law of uncertainty propagation with a numerical technique similar to the Monte Carlo method. Since the inputs of the algorithm are images, a model of the camera noise representing input uncertainties, is designed. Both uncertainty assessment techniques provide comparable results.

Many authors evaluated the uncertainty of edge detection algorithms such as Canny [91] operator or Laplacian of Gaussian operator [92]. In the first paper, three different sources of uncertainty are chosen: illumination changes, vibrations and image quantization. Considering these impacting sources, some noise is added to the pixels to obtain the uncertainty of a point being part of an edge. Basically, a distribution probability is associated to each point of the image from which a standard deviation can be computed. The mathematical development of the detector is shown, hence the use of the law of uncertainty propagation [91]. Similar work is done in [92] excepting that the modeling of the uncertainty of image pixels is slightly different.

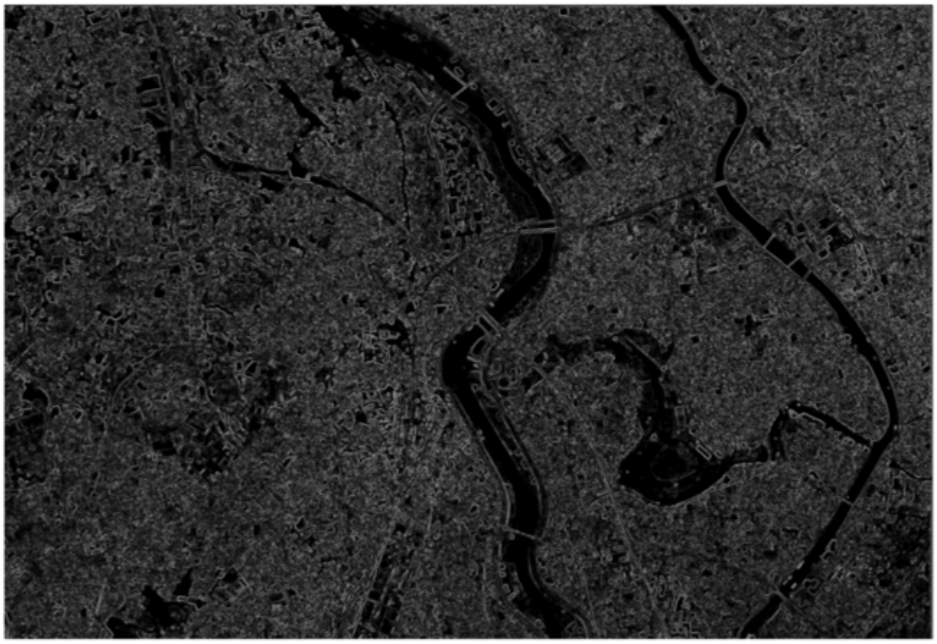
Different statistical, structural and filter-based techniques are studied and their associated uncertainty is computed in [93]. They reviewed some common methods like thresholding, histogram equalization, several edge detection operators, different filters, etc. Different noise modelings are proposed as input uncertainties such as Gaussian noise or Salt and Pepper. The uncertainty propagation law is the assessment method used. The authors also provided a clear manner to visualize the propagated uncertainty directly on the image as illustrated in figure 2.10. This figure depicts a map of Tokyo (a) and the output image of the edge detection algorithm (b). White areas represent certain edges while gray and orange ones are medium to high uncertainty zones.

The uncertainty of the Discrete Fourier Transform algorithm is evaluated in [94] using the method described in the GUM [13]. Even though the paper is not focused on digital images processing examples, it describes the mathematical background of this ubiquitous technique. One of the conclusions drawn is that quantization is the most impacting uncertainty source which is also present in image-based measurements.

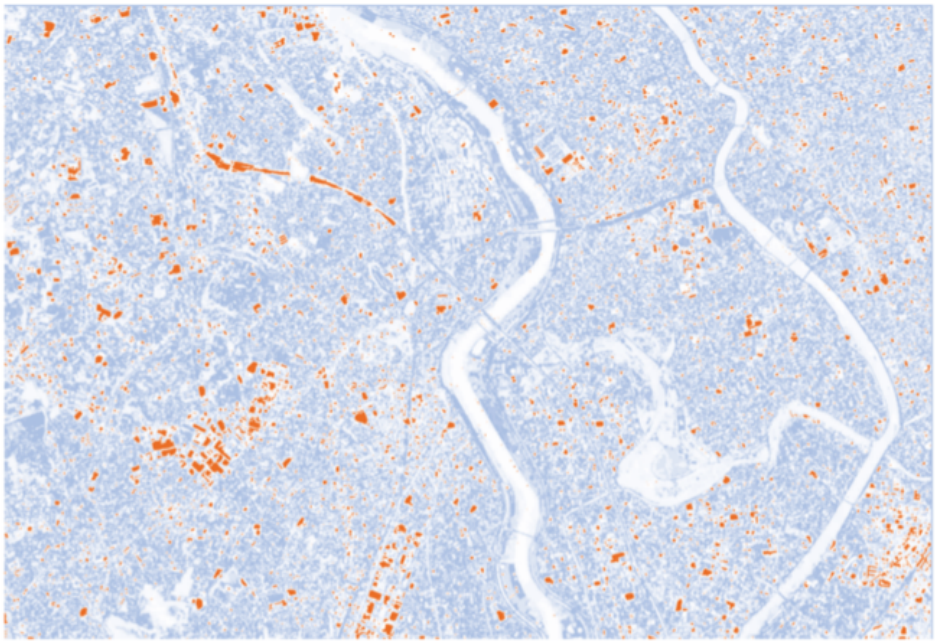
Some other methods used to assess the uncertainty in computer vision algorithms do exist such as Dempster-Shafer belief theory and fuzzy set theory [95]. The properties of the images such as the intensity can be represented by a fuzzy set. The uncertainty propagation is carried out by defining for each pixel intensity a membership function which is an interval with a range from 0 to 1 [95]. The membership degree for a pixel represents how well it is a member of the fuzzy set [96]. In addition to handle uncertain knowledge, fuzzy theory can also be used for optimization problems [97, 98]. In Dempster-Shafer belief models the uncertainty is based on a mass function from which a belief function and a plausibility function can be derived rather than density probability function as in the Monte Carlo method [99]. These two methods are not discussed in this literature review since the implementation of them requires more substantial modifications to AV&R's algorithms than the Monte Carlo method.

2.3 Review conclusion

This project involves three common studied fields: the calibration of a camera, the detection of defects and the uncertainty evaluation of a model. Although each of these domains is the subject of many papers, it is difficult to find one that deals with the three topics, i.e. the evaluation of the uncertainties of a defect detection algorithm and the associated images acquired by a calibrated camera. The aim of this work is therefore to deepen the knowledge of the uncertainty quantification of visual inspection systems. Even though this project is mainly focused on uncertainties rather than the development of new calibration techniques and detection algorithms, the implementation of a planar camera calibration is carried out as it could improve the results obtained by AV&R. Conventional image processing techniques were developed by AV&R for the detection and their algorithms are analyzed during this work in a black box way for confidential purposes. The Monte Carlo method is chosen to be adequate for this assessment since all the models involved in this project are highly non-linear. For the calibration methods, type B uncertainty evaluation is used since information on distribution and input uncertainties is available. For the detection algorithms, type B evaluations are mainly used except for the uncertainty estimation of the calibrated images for which a series of independent observations was used (type A).



a)



b)

Figure 2.10 Uncertainty visualization : a) input image of Tokyo map b) edge detection with associated uncertainty reproduced with permission from [93]

CHAPTER 3 RATIONALE OF THE PROJECT

3.1 Summary of the problem

This project focuses on the uncertainty evaluation of the visual inspection system of AV&R. The inspection process developed by AV&R in close collaboration with Rolls-Royce Canada Limited contains a lot of complex steps as depicted in figure 3.1. The end goal of AV&R is to be able to assess the uncertainty of their whole visual maintenance system and conduct a sensitivity analysis to identify which steps have a higher impact on the defect measurement uncertainty. Since the inspection routine is composed of a high number of uncertainty sources, it was decided to rank the three subsystems that seemed the most impacting. Therefore the schematic depicted in figure 3.1 was drawn with the help of AV&R and Rolls-Royce Canada Limited to highlight those three impacting systems based on their experience. It resulted that the three most important systems are the 2D and 3D acquisitions processes and the calibration of the camera and the scanner. Precisely, the study of the 2D acquisition and the calibration of the camera were chosen to be the core of this thesis.

Before the calibration of external parameters, the focus and the depth of field are adjusted. In this work, the uncertainty of the focus and depth of field calibration method can only be evaluated using numerical methods since the model is nonlinear and thus analytical methods cannot be applied. The external parameters calibration cannot be completely implemented due to technical and budget constraints. Hence, a pose estimation algorithm (POSIT) is implemented on a similar setup in the CoSIM Lab and the associated uncertainties can also be quantified by using numerical methods.

Regarding the detection routine, one sensitive deterministic edge detection algorithm, *Slope detection* is mainly taken into account throughout this work due to the time constraints of a master. AV&R wants to quantify the sensitivity of the detection outputs of this algorithm to the variation of three key parameters namely a threshold, the size of a filter and the minimum size of detected defects. For confidentiality reasons, the analysis of the detection routine is done in a black box way. Furthermore, the detection uncertainty of the *Slope Detection* algorithm is carried out. Hence, numerical methods such as the Monte Carlo method are appropriate.

The uncertainty assessment of the 2D acquisition process and the associated calibration and detection techniques allows AV&R to have a better insight into the quality of their measurement system. Indeed, knowing the uncertainty of AV&R's algorithms outputs will

help define an interval within which the expected measurements lie and thus check that the defined tolerances are still respected. Furthermore, it is interesting for AV&R to be sure that their high-precision measuring system contains low uncertainties or to be aware of them in order to mitigate them.

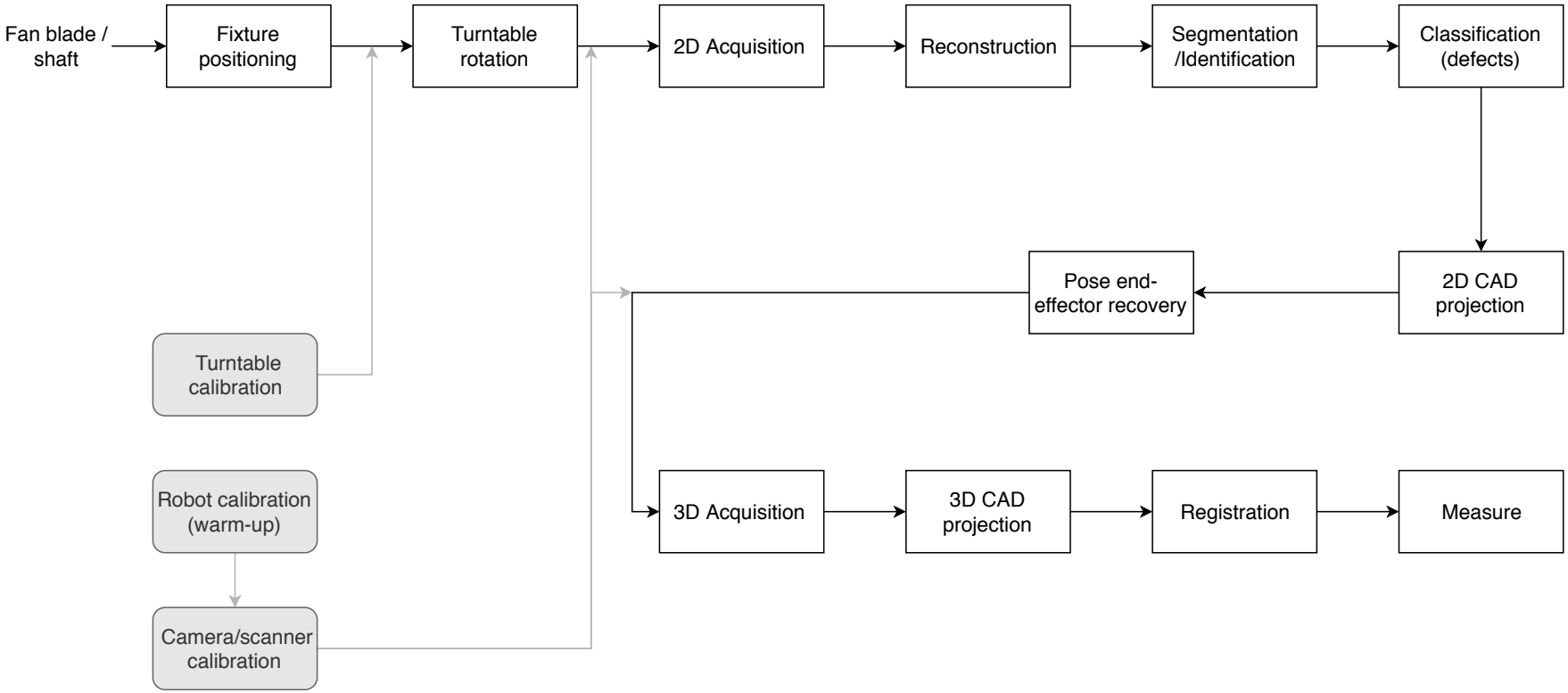


Figure 3.1 Inspection process diagram

3.2 General objective

The main objective of this project is to quantify the uncertainty of the calibration processes and defect detection algorithms for one critical aeronautical part: a fan blade. Especially the critical area: the zone close to the root and the leading edge.

3.3 Specific objectives

- Specific objective 1 (S.O.1): quantify the uncertainty of the focus calibration process developed by AV&R
- Specific objective 2 (S.O.2): implement a pose estimation algorithm and quantify the uncertainty of the camera pose
- Specific objective 3 (S.O.3): quantify the impact of the surface finish and the defect location on the uncertainty of the detection algorithms

3.4 Hypothesis and scope

As illustrated in figure 3.1, the 2D acquisition and its calibration require preliminary steps such as the positioning of the part in the fixture and the turntable calibration. These two processes contain different sources of uncertainty that propagate to the acquisition. The first assumption made is that these propagated uncertainties are not considered in the evaluation of the acquisition uncertainty.

Even though the camera is held by the robot at the end effector, the hand-eye configuration is not considered in this project for the following reason. The mechanical uncertainty introduced by the robot displacement has an impact on the measurement uncertainty of the defects. Since this uncertainty has been studied by AV&R, it is not included in this work. However, once all uncertainties related to the fixture positioning and the turntable are evaluated, a sensitivity analysis could be carried out to highlight the relative uncertainty of each process.

As mentioned there are lots of detection algorithms used in the whole detection routine, we decided to evaluate the uncertainties of one sensitive algorithm. For confidentiality purposes, the studied detection algorithm is considered as a black box.

The following hypotheses are stated here and tested in section 5.

- The pixels of the calibrated images present an uncertainty significantly high to notice a non zero uncertainty of the detection outputs

- The detection uncertainty is not constant in the critical zone of the fan blade and depends on the lighting direction

Several limitations slowed down the progress or the depth of the results obtained during this project. Concerning the calibration methods of AV&R, intermediate images (debug images) and measurements or results, useful when adapting an algorithm, cannot be easily extracted. Therefore, the results obtained when implementing those methods can hardly be compared with the original ones.

Intrinsic parameters are assumed to be fixed for AV&R, thus a complete camera calibration cannot be implemented. Furthermore, the camera reference frame is not defined at its optical center in AV&R's algorithms as it is the case in the literature. The homogeneous transformation matrix between those two frames position is required to compare the results if the proposed calibration method (POSIT) is implemented onto AV&R's setup.

Regarding the detection routine, the computing time required to inspect the fan blade is too high to run 200,000 simulations as recommended by the GUM [13]. It would last about 20 days. Therefore, the number of simulations is limited to 100 or 1,000 depending on the experiment. The computation time is thus reduced to 10h for 100 simulations and 100h for 1,000. Finally, the three aforementioned key parameters of the *Slope Detection* algorithm need to be modified by hand for each iteration if one wants run random combinations which is tiresome for thousands of iterations. Finally, the work presented hereunder corresponds to one version of the detection routine and does not take the modifications that have been made during the project.

CHAPTER 4 METHODOLOGY

4.1 Calibration methods

For the sake of confidentiality, calibration algorithms developed by AV&R are not detailed in this section. Nevertheless, a brief overview of their working principle is presented. Since both methods were initially developed on AV&R's software, they were adapted to a suitable programming software: MATLAB. Results of implemented methods are compared with original ones to assess the validity of the implementation before evaluating the uncertainty of the output parameters. The other two calibration techniques coming from the literature were also implemented in MATLAB. Those methods are detailed hereunder.

4.1.1 Depth of field and focus

In the calibration methods mentioned in section 2.1.1, the camera is considered focused (called also in focus). Thus, before computing intrinsic and extrinsic parameters, the focus needs to be adjusted. The focus position u is the sharpest point in an image while the depth of field (DOF) is the zone where the image is still considered in focus [100]. As depicted in figure 4.1, the depth of field is defined by the near limit of focus u_n and the far limit of focus u_f. The depth of field can be computed using the circle of confusion of radius ρ. The lens diameter D and the lens thickness t are also represented in the figure 4.1. In a complete automated system, there exist autofocus algorithms like the one described in [100]. In this project, the calibration of the focus requires manual operations.

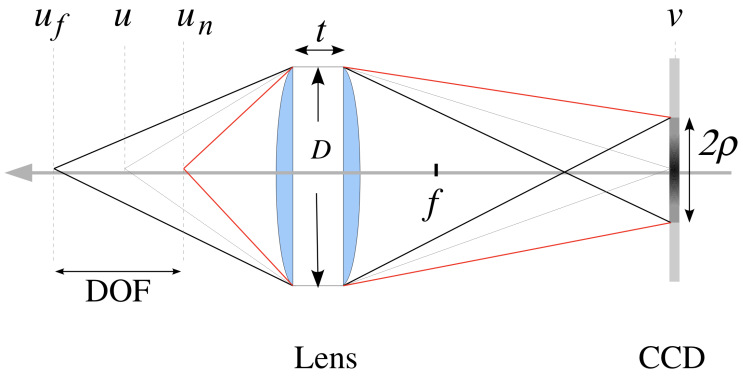


Figure 4.1 Schematic of the in-focus position u and the depth of field limits reproduced with permission from [100] © 2014 Springer Science Business Media New York

The calibration target used by AV&R to calibrate the focus and the depth of field is shown in figure 4.2. The region of interest corresponds to the vertical lines. The image reference frame is placed at the top left corner with the x-axis pointing to the right and the y-axis pointing downward. The idea is to find the pixel with the highest intensity on those lines, i.e. the focus point. This point is then defined as the center of the depth of field segment along the y-axis.

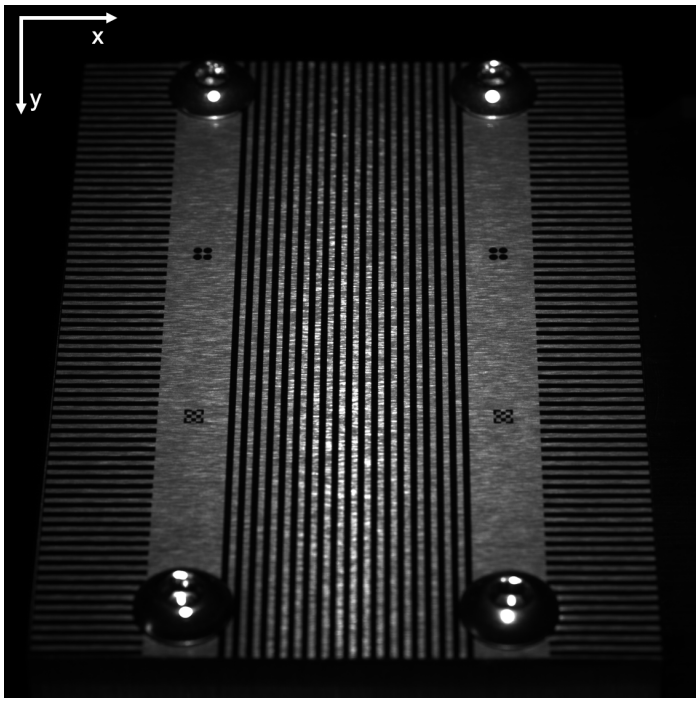


Figure 4.2 Target used for the calibration of the focus/DOF © 2019 AV&R

An overview of the different steps of this calibration algorithm is depicted in figure 4.3. The core of the method lies in the identification of the spline parameters that fit best the average intensity value of each row. The top of the spline corresponds to the focus point. Finally, the validation step ensures that obtained results follow specifications of AV&R. Indeed, even though there is an optimal value to reach for each output parameter, a tolerance zone around each optimum is defined by AV&R. For the focus position the target is 1,024 pixels while the tolerance is 50 pixels. For the depth of field, the target is 340 pixels and the tolerance is also 50 pixels. The implementation in MATLAB uses the same type of spline and the same spline smoothness parameter.

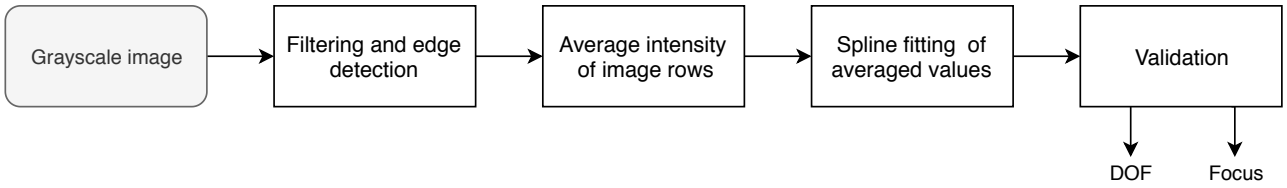


Figure 4.3 Working principle of the calibration of the focus/DOF

4.1.2 Extrinsic parameters

Once focused, the camera parameters can be computed. The method developed by AV&R to determine the optical configuration of the camera is not studied in this project. Hence, intrinsic parameters are considered to be fixed and unknown. Nevertheless, the pose of the camera (extrinsic parameters) can be estimated. As aforementioned, the camera is located at the end effector of an articulated robot. The calibration gauge is composed of 4 circles manufactured with precision and measured with a Coordinate Measuring Machine (CMM) (figure 4.4). Actually, this calibration method uses 5 control points : the centers of the circles and the center of the square they form. To determine the external parameters, the camera is translated along its optical axis while measurements are taken.

First, filtering and thresholding algorithms are used to obtain a binary image with as less artifacts as possible. Knowing the approximate size of the circles on the image, they can be identified using blob detection techniques [101]. Once detected, their centroid are determined and the center of the square they form is computed using lines intersection formulas. To proceed to the estimation of the translation vector and the three rotations angles, different measurements of the 5 control points are taken at several distances from the calibration gauge.

The calibration method used by AV&R is implemented but some information is missing in order to be able to validate the results. Indeed, the implementation in MATLAB is done without considering the displacement of the robot holding the camera. The images taken at each measurement position are thus required to complete the calibration method. The extraction of those images is not straightforward with the actual software of AV&R. Since the implementation of this extraction tool could take several weeks considering the actual pipeline of the project, an alternative calibration method, the POSIT algorithm developed in [42], is proposed but cannot be directly implemented onto AV&R's setup for two reasons. Firstly,

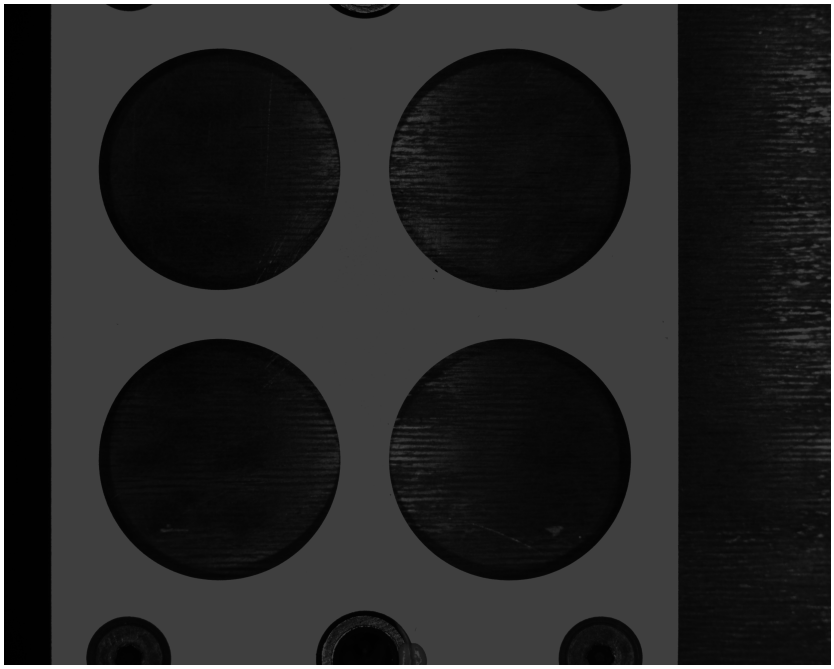
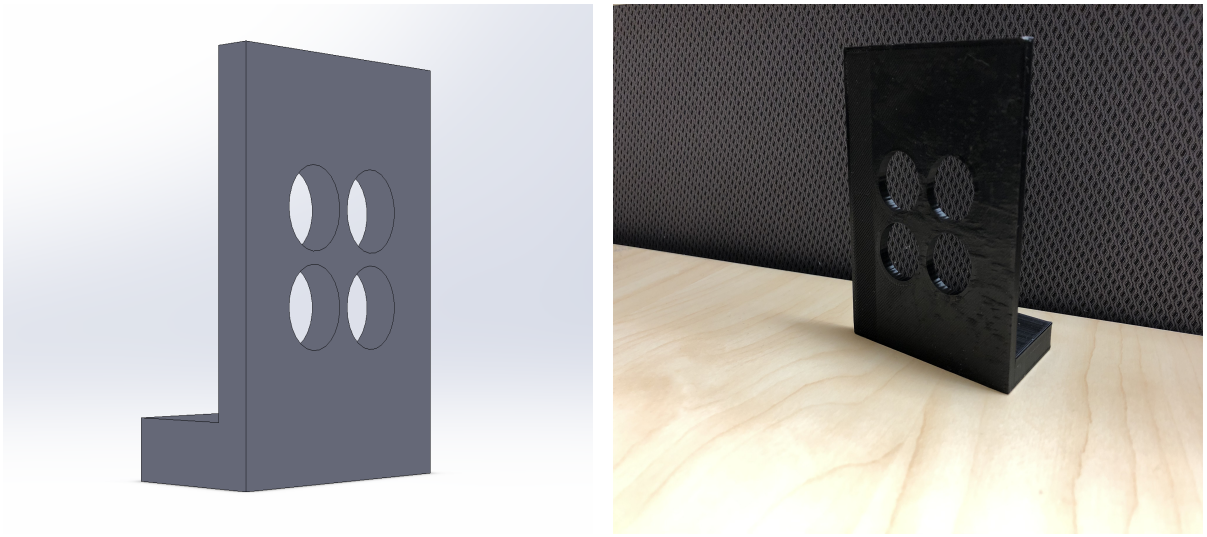


Figure 4.4 Target used to calibrate the external parameters of the camera © 2019 AV&R

the implementation of the POSIT algorithm requires modifications of the setup leading to extra cost considered too high by the company. Secondly, in AV&R's calibration method the camera reference frame is not defined at its optical center as it is the case in the pinhole model. The homogeneous transformation matrix between those two frames is not known yet. Thus, even if the POSIT algorithm was implemented on the setup, the results could not be compared with the actual calibration of AV&R since the rotation matrix and the translation vector are not defined in the same reference frame.

To overcome these obstacles and quantify the accuracy and the uncertainty of the proposed calibration method, a part of AV&R calibration setup is reproduced. For that purpose, a similar calibration target is designed on SolidWorks and then 3D printed using a MakerGear M2 in the CoSIM lab of Pr. Achiche. The CAD of this target and the printed part are shown in figure 4.5. The diameter of the circles is 20 mm and their center are 23 mm apart.

AV&R uses an industrial camera for its accuracy. Unfortunately, such an expensive camera cannot be purchased for this work. Nevertheless, a USB camera with low distortion and a high resolution is ideal for the POSIT algorithm. The characteristics of the purchased camera See3CAM CU135 are summarized in table 4.1. In our setup, the camera is not mounted on a robot but placed at fixed and known positions. An overview of the setup is presented in figure 4.6. Since the intrinsic parameters are assumed to be known in the POSIT algorithm [42], we need to estimate them with another method. Therefore, we choose one of the



(a) CAD (b) 3D printing

Figure 4.5 Designed circular calibration target

most implemented method for its accuracy : Zhang's method [38]. Even though, the image resolution can be as large as stated in table 4.1, we use a resolution of 1280x720 to avoid an image capture time of several seconds leading to motion blur in the images.

Table 4.1 Characteristics of the USB camera See3CAM CU135

Parameter	Units	Value
Focal length f	mm	4.3
F-number	/	$f/2.8$
Sensor size	inch	1/3.2
Distortion	%	<-1.2
Image resolution	pixels	4208x3120
Total price	CAD	359.08

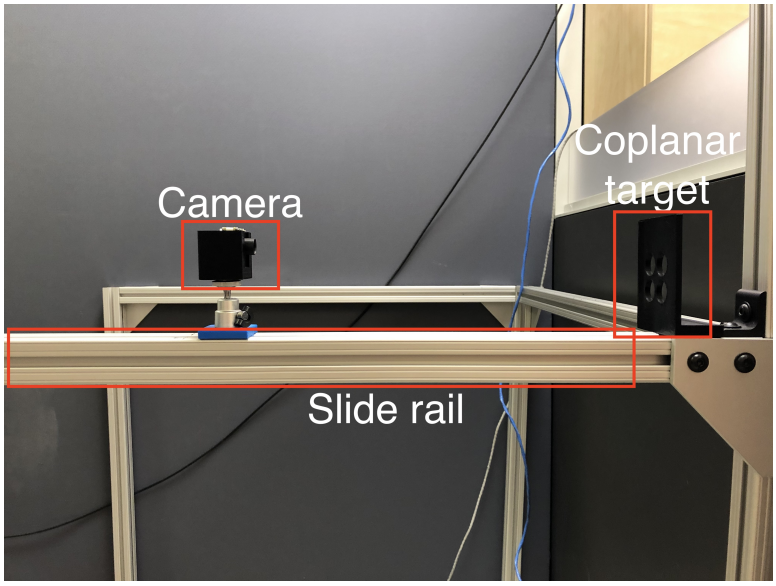


Figure 4.6 Setup of the camera calibration (CoSIM lab of Pr. Achiche)

Zhang's method

Zhang's method is directly implemented in MATLAB. Nevertheless, the principle is presented hereunder [38]. Since this calibration technique requires more than four points to provide consistent results, a planar target with 32 squares of 23 mm side is printed and used. As mentioned in section 2.1.1, the goal of the calibration is to determine the intrinsic matrix $\mathbf{K}$ and the pose of the camera $[\mathbf{R} \mid \mathbf{t}]$ (equation 4.1). Each 3D point is represented by a vector $\mathbf{M} = [X_i, Y_i, Z_i]^T$ while its image is a 2D point $\mathbf{m} = [u_i, v_i]^T$. The definition of each internal parameter is described in table 2.1.

$$\lambda \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \mathbf{K} \begin{bmatrix} \mathbf{R} & | & \mathbf{t} \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix} \tag{4.1}$$

To simplify the problem, the target is considered to be at Z = 0; on the X-Y plane of the *World* reference frame. Each element of the third column of the rotation matrix $\mathbf{R}$ is thus multiplied by 0. If the remaining columns are respectively denoted $\mathbf{r_1}$ and $\mathbf{r_2}$, a homography $\mathbf{H}$ can be defined up to a scale factor λ between the target plane and its image (equation

4.2).

$$\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \mathbf{H} \begin{bmatrix} X_i \\ Y_i \\ 0 \\ 1 \end{bmatrix} \tag{4.2}$$

This homography is thus equal to the camera parameters matrix (equation 4.3). Each column of the homography matrix is written as $\mathbf{h_i}$.

$$\mathbf{H} = [\mathbf{h_1}\ \mathbf{h_2}\ \mathbf{h_3}] = \lambda \mathbf{K}[\mathbf{r_1}\ \mathbf{r_2}\ \mathbf{t}] \tag{4.3}$$

Two constraints can be specified based on the orthonormality of vectors of a rotation matrix (equation 4.4). A geometric interpretation of those constraints is described in [38].

$$\begin{cases} \mathbf{h_1^T K^{-T} K^{-1} h_2} = 0 \\ \mathbf{h_1^T K^{-T} K^{-1} h_1} = \mathbf{h_2^T K^{-T} K^{-1} h_2} \end{cases} \tag{4.4}$$

The matrix $\mathbf{B}$ can be defined as

$$\mathbf{B} = \mathbf{K^{-T} K^{-1}} = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix} \tag{4.5}$$

Since $\mathbf{B}$ is symmetric, only 6 components are required to build the entire matrix (equation 4.6). By defining $\mathbf{v_{ij}}$ as followed, we can write the equation 4.7.

$$\begin{cases} \mathbf{b} = [B_{11}, B_{12}, B_{22}, B_{13}, B_{23}, B_{33}]^T \\ \mathbf{v_{ij}} = [h_{i1}h_{j1}, h_{i1}h_{j2} + h_{i2}h_{j1}, h_{i2}h_{j2}, h_{i3}h_{j1} + h_{i1}h_{j3}, h_{i3}h_{j2} + h_{i2}h_{j3}, h_{i3}h_{j3}]^T \end{cases} \tag{4.6}$$

$$\mathbf{h_i^T B h_j} = \mathbf{v_{ij}^T b} \tag{4.7}$$

We can rewrite both constraints on the intrinsic parameters as :

$$\underbrace{\begin{bmatrix} \mathbf{v_{12}^T} \\ (\mathbf{v_{11}} - \mathbf{v_{22}})^\mathbf{T} \end{bmatrix}}_{\mathbf{V}} \mathbf{b} = 0 \tag{4.8}$$

For n points of views or images of the object plane, one has $2n$ rows in the matrix $\mathbf{V}$. Thus,

a minimum number of 3 images is required to obtain a unique solution.

By solving this system of equations, we find $\mathbf{b}$ hence $\mathbf{B}$. Once $\mathbf{B}$ is determined, the intrinsic parameters can be easily estimated (equation 4.9). The definition of the intrinsic parameters are shown in table 2.1.

$$\begin{cases} v_c = \frac{(B_{12}B_{13} - B_{11}B_{23})}{(B_{11}B_{22} - B_{12}^2)} \\ \lambda = B_{33} - \frac{[B_{13}^2 + v_c(B_{12}B_{13} - B_{11}B_{23})]}{B_{11}} \\ \gamma = c_u f = \sqrt{\frac{\lambda}{B_{11}}} \\ \eta = c_v f = \sqrt{\frac{\lambda B_{11}}{B_{11}B_{22} - B_{12}^2}} \\ \alpha = \frac{-B_{12}\gamma^2\eta}{\lambda} \\ u_c = \frac{\alpha v_c}{\gamma} - \frac{B_{13}\gamma^2}{\lambda} \end{cases} \tag{4.9}$$

When the intrinsic matrix is computed, the pose of the camera can be obtained using equation 4.10.

$$\begin{cases} \mathbf{r_1} = \lambda \mathbf{K^{-1}} \mathbf{h_1} \\ \mathbf{r_2} = \lambda \mathbf{K^{-1}} \mathbf{h_2} \\ \mathbf{r_3} = \mathbf{r_1} \times \mathbf{r_2} \\ \mathbf{t} = \lambda \mathbf{K^{-1}} \mathbf{h_3} \\ \lambda = \frac{1}{||\mathbf{K^{-1}} \mathbf{h_1}||} = \frac{1}{||\mathbf{K^{-1}} \mathbf{h_2}||} \end{cases} \tag{4.10}$$

It is important to note that due to noise, the properties of the obtained matrix $[\mathbf{r_1}\ \mathbf{r_2}\ \mathbf{r_3}]$ dot not fully fulfill the properties of a rotation matrix. Nevertheless, it is possible to retrieve a rotation matrix from this 3x3 matrix corrupted by noise [38]. Radial distortion is then added to Zhang's model once the previous closed-form solution is computed (equation 4.11). The distorted point coordinates and the ideal projected point coordinates are respectively denoted (X_d, Y_d) and (X_u, Y_u). As described in the literature review, r is the radial distance from the principal point (u_c, v_c) and κ_1 and κ_2 are the radial distortion coefficients. The total radial distortion of the x and y coordinates is respectively denoted δ_x and δ_y. Furthermore, the Camera Calibrator App of MATLAB [102] allows to add tangential distortion coefficients p_1 and p_2 to the initial distorted model.

$$\begin{cases} X_d = X_u + \delta_x = X_u(1 + \kappa_1 r^2 + \kappa_2 r^4) \\ Y_d = Y_u + \delta_y = Y_u(1 + \kappa_1 r^2 + \kappa_2 r^4) \\ r = \sqrt{X_d^2 + Y_d^2} \end{cases} \tag{4.11}$$

To refine the solution, the maximum likelihood criterion is used. Considering m control

points on n different points of views, the following objective function can be minimized (equation 4.12) with $\hat{m}(K, \kappa_1, \kappa_2, , p_1, p_2, R_i, t_i, M_j)$ being the perspective projection of a 3D point M_j while m_{ij} is the associated 2D point observed on the image. The minimization algorithm integrated into MATLAB is Levenberg-Marquardt [38].

$$\sum_{i=1}^{n}\sum_{j=1}^{m} ||m_{ij} - \hat{m}(K, \kappa_1, \kappa_2, p_1, p_2, R_i, t_i, M_j)||^2 \tag{4.12}$$

52 images with different orientations are acquired with 49 control points per image. The algorithm returns the intrinsic matrix $\mathbf{K}$, the radial distortion coefficients κ_1 and κ_2, the tangential distortion coefficients p_1 and p_2, the pose for each target position and the reprojection error, i.e the error between an image point and its projection. Among all these outputs, the focal length and the image center will be used for the POSIT algorithm.

POSIT algorithm

The Pose from Orthography and Scaling with Iterations (POSIT) algorithm consists in estimating a pose of the camera iteratively from a single image using an orthographic and scaled projection model [42]. In [42], this algorithm is applied for the particular case of coplanar points, which is a degenerate case when a perspective projection model is used [26]. The scaled orthographic projection is adequate when the distance between the camera and the object is large compared to the object depth. Therefore, control points with different depth coordinates are seen as coplanar. Moreover, the scaled orthographic projection model provides two valid solutions for the estimated pose as depicted in figure 4.7. These possible poses are mirrored with respect to a plane parallel to the image plane. For each iteration, the algorithm returns thus two solutions. Any non acceptable pose returned by the algorithm is discarded. This situation could happen if the z coordinate of a point is negative, i.e. being behind the camera. In the case of two valid poses, the optimal one is chosen by computing the projection error.

The parameters used in this method and their description are summarized in table 4.2. As mentioned, the focal length and the image center are assumed to be known and are computed, in this work, by means of Zhang's method [38]. The figure 4.8 shows the representation of the scaled orthographic projection p_i of a 3D point M_i and the perspective projection m_i of the same point. The image point p_i of a non coplanar 3D point M_i is obtained by using an orthographic projection P_i on the plane D and then a perspective projection on the plane G. A correction factor ϵ_i between the scale orthographic projection and the perspective projection is defined for each point. Since the control points of the calibration target used (figure

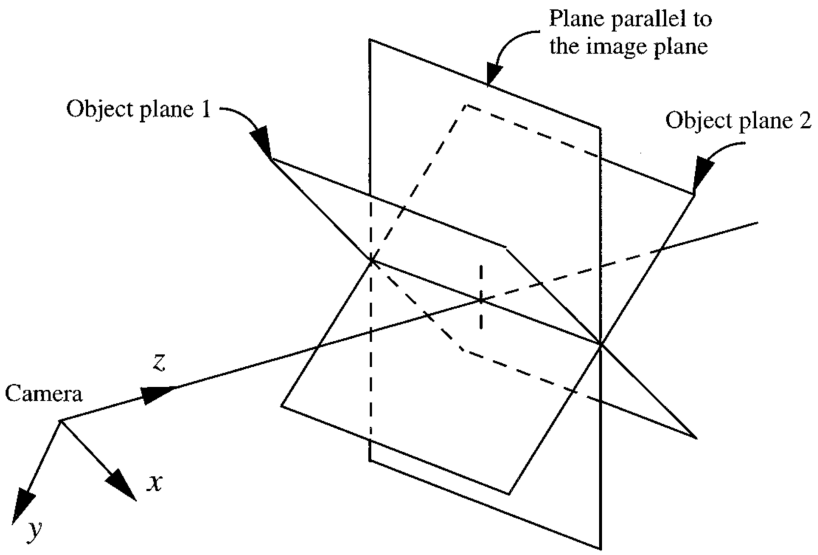


Figure 4.7 Double mirrored solutions for the POSIT algorithm reproduced with permission from [42] © 1996 Academic Press

4.5) are ideally coplanar, ϵ_i is close to 0 and M_i and P_i almost coincide on the plane D.

Table 4.2 Notation of POSIT parameters

Parameter	State	Description
f	Known	Focal length
$\mathrm{C} = (u_c, v_c)$	Known	Image center
$\mathbf{P_i^W} = [X_i, Y_i, Z_i]$	Known	Control points defined in the *World* reference frame
$\mathbf{p_i} = [u_i, v_i]$	Known	Image of the control points on the image plane G
$\mathbf{T} = [t_x, t_y, t_z]^T$	Unknown	Translation vector
$\mathbf{R} = [\mathbf{i}, \mathbf{j}, \mathbf{k}]^T$	Unknown	Rotation matrix
$\mathbf{P_i^C} = [x_i, y_i, z_i]$	Unknown	Control points defined in the camera reference frame
$\epsilon_\mathbf{i}$	Unknown	Correction factor vector between perspective projection points and scaled orthographic points

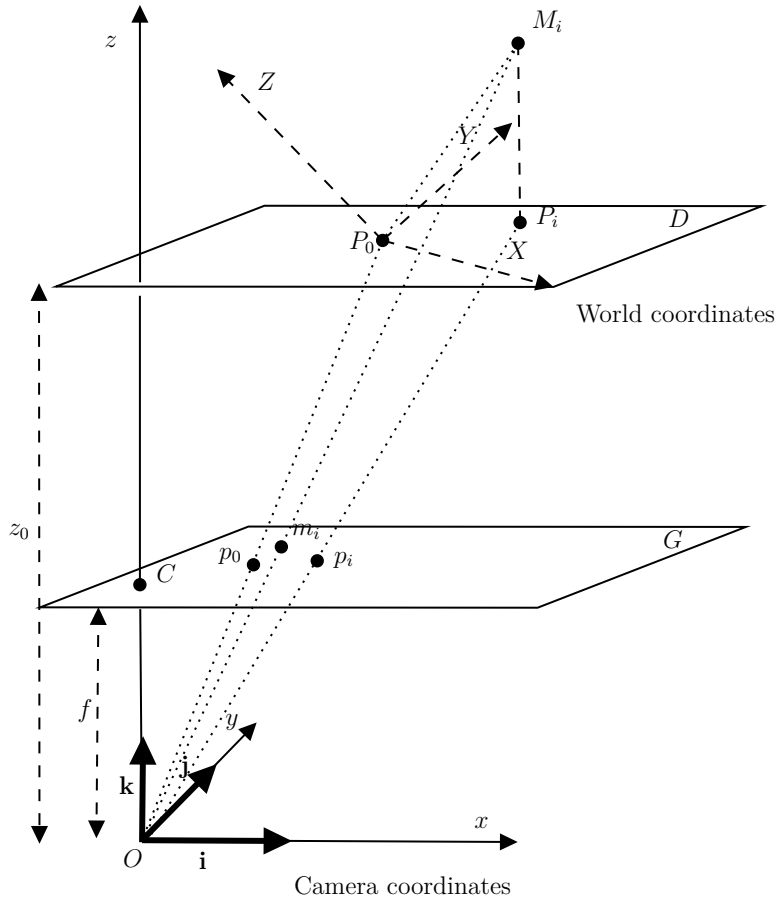


Figure 4.8 Schematic representation of the scaled orthographic projection and the corresponding perspective projection reproduced with permission from [42] © 1996 Academic Press

The goal of the algorithm is to find the 4x4 homogeneous transformation matrix **S** between the *World* reference frame and the camera reference frame, i.e. the pose of the camera. This matrix is defined in equation 4.13. It is composed of a rotation matrix **R** and a translation vector **T**. The unit vectors **i**, **j** and **k** of the rotation matrix **R** are column vectors.

$$\begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} = \mathbf{S} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{i}^T & t_x \\ \mathbf{j}^T & t_y \\ \mathbf{k}^T & t_z \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix} \tag{4.13}$$

Two of the three unit vectors of **R** need to be computed to define completely the rotation matrix since the vectors are orthogonal. Vectors **i** and **j** are thus unknown parameters while **k** is obtained by taking the cross-product of those vectors. The origin of the *World* reference

frame is defined as $P_0 = [x_0, y_0, z_0]$ in the camera reference frame which corresponds to the translation vector $\mathbf{T}$. If no distortion is considered, we can define :

$$\begin{cases} u_i = \frac{fx_i}{z_i} \\ v_i = \frac{fy_i}{z_i} \end{cases} \tag{4.14}$$

Considering first a non coplanar case, the terms of equations 4.14 can be rearranged using equation 4.13 to obtain :

$$\begin{cases} [X_i, Y_i, Z_i] \cdot \mathbf{I} = u_i(1 + \epsilon_i) - u_0 \\ [X_i, Y_i, Z_i] \cdot \mathbf{J} = v_i(1 + \epsilon_i) - v_0 \end{cases} \tag{4.15}$$

With

$$\begin{cases} \mathbf{I} = \frac{f}{z_0}\mathbf{i} \\ \mathbf{J} = \frac{f}{z_0}\mathbf{j} \\ \epsilon_\mathbf{i} = \frac{1}{z_0}[X_i, Y_i, Z_i] \cdot \mathbf{k} \end{cases} \tag{4.16}$$

Without considering the correction factor vector $\epsilon_\mathbf{i}$, we need to determine 5 parameters to compute the pose of the camera : $\mathbf{i}, \mathbf{j}, u_0, v_0$ and z_0. When all control points are perfectly coplanar, $\epsilon_\mathbf{i}$ is a vector full of zeros. To compute the value of $\mathbf{I}$ and $\mathbf{J}$, we solve a linear system of equations (equation 4.17). $\mathbf{B}$ is the pseudoinverse of the matrix containing all the 3D points $[X_i, Y_i, Z_i]$ and $\mathbf{u}$ and $\mathbf{v}$ are the vectors containing the right hand side of the equations 4.15.

$$\begin{cases} \mathbf{I} = \mathbf{Bu} \\ \mathbf{J} = \mathbf{Bv} \end{cases} \tag{4.17}$$

The solution of equation 4.17 is denoted $\mathbf{I_0}$ and $\mathbf{J_0}$ when points are coplanar. We define the vector $\mathbf{w}$, a unit vector normal to the object plane D. Hence, $[X_i, Y_i, Z_i].\mathbf{w} = 0$. The coplanar solution is defined in equation 4.18 with ϕ and μ being the coordinate of the head of $\mathbf{I}$ and $\mathbf{J}$ along $\mathbf{w}$. A detailed development of this solution can be found in Appendix A.

$$\begin{cases} \mathbf{I} = \mathbf{I_0} + \phi\mathbf{w} \\ \mathbf{J} = \mathbf{J_0} + \mu\mathbf{w} \end{cases} \tag{4.18}$$

In our case, the matrix containing all the 3D points is not full rank. Hence, other constraints are required. Two other constraints come from the fact that $\mathbf{I}$ and $\mathbf{J}$ must have the same

norm and must be perpendicular (equation 4.19).

$$\begin{cases} \phi\mu = -\mathbf{I_0} \cdot \mathbf{J_0} \\ \phi^2 - \mu^2 = ||\mathbf{J_0}||^2 - ||\mathbf{I_0}||^2 \end{cases} \tag{4.19}$$

To find a solution for ϕ and μ, we can define a complex number C and its squared form (equation 4.20)

$$\begin{cases} C = \phi + i\mu \\ C^2 = \phi^2 - \mu^2 + 2i\phi\mu = \rho e^{i\theta} \end{cases} \tag{4.20}$$

By using equation 4.19 in equation 4.20, we obtain:

$$C^2 = ||\mathbf{J_0}||^2 - ||\mathbf{I_0}||^2 - 2i\mathbf{I_0} \cdot \mathbf{J_0} \tag{4.21}$$

By computing the real and imaginary parts of C, the value of ϕ and μ is determined (equation 4.22). Therefore, there are two solutions for $\mathbf{I}$ and $\mathbf{J}$ due to the square root and thus two possible poses. These mirror solutions are the ones shown in figure 4.7.

$$\begin{cases} \phi = \pm\sqrt{\rho}cos(\theta/2) \\ \mu = \pm\sqrt{\rho}sin(\theta/2) \end{cases} \tag{4.22}$$

Since for each pose iteration, two rotation matrices are computed we need to check if both poses are acceptable. If the pose presents at least one point with z_i <0, it is an erroneous pose. Indeed, a negative z coordinate would mean that the 3D point is located behind the camera. Once $\mathbf{I}$ and $\mathbf{J}$ are determined, z_0 can be estimated using the norm of $\mathbf{I}$ or $\mathbf{J}$ (equation 4.16). Knowing the image of the reference point P_0, x_0 and y_0 can be computed by means of the equation 4.14. The figure 4.9 summarizes the working principle of the POSIT algorithm.

The error E is the average Euclidean distance defined in equation 4.23 with (u_i, v_i) being the image of the control points and (u_i^p, v_i^p) the projected image points. There are m control points.

$$E = \sum_{i=1}^{m} \frac{1}{m} \sqrt{(u_i - u_i^p)^2 + (v_i - v_i^p)^2} \tag{4.23}$$

The definition of the target reference frame used and the 3D control points are presented in Appendix B.

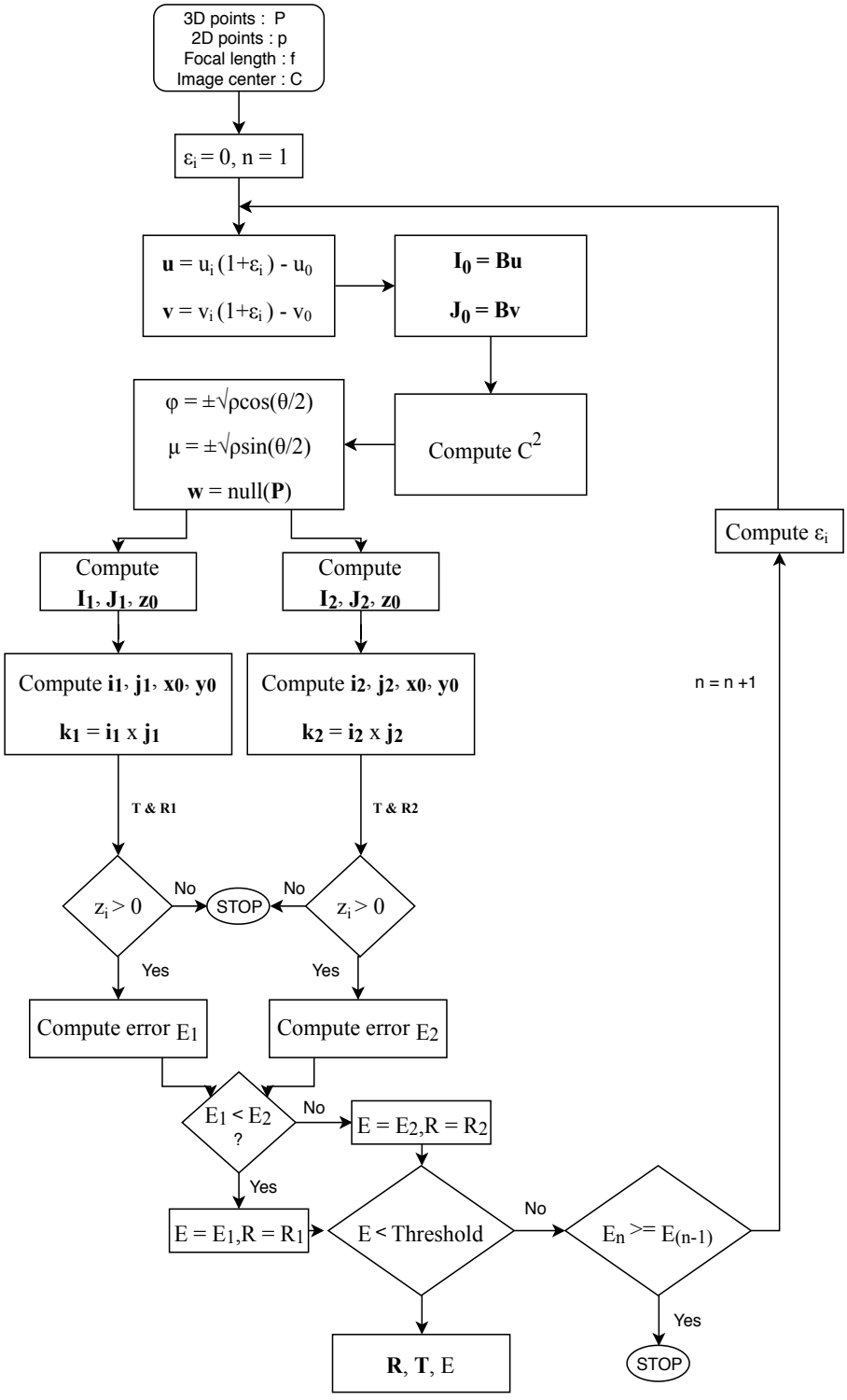


Figure 4.9 Working principle of the POSIT algorithm reproduced with permission from [42] © 1996 Academic Press

4.2 Detection routine

The detection routine developed by AV&R is composed of several combinations of image processing algorithms that vary depending on the inspected zone of the part. The most critical area of the fan blade being studied throughout this project is located near the root and the leading edge and it is depicted in figure 4.10. To cover the entire zone, 8 different acquisitions are taken. These acquisitions are represented for illustrative purposes but they do not have the proper scale. It is important to state that this work is based on a particular version of the detection routine and does not take into account all the updates and modifications that have taken place during the master.

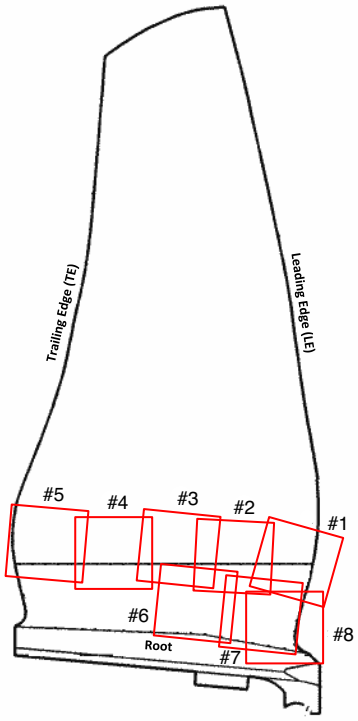


Figure 4.10 Schematic of a fan blade and the acquired acquisitions of the fan blade critical zone © 2019 AV&R and Rolls-Royce Canada Limited

The inspection batch contains 20 different parts. Those blades have been repaired in the past but contain lots of defects. Since one iteration of the whole routine lasts 12 minutes for those 20 fan blades, we decided to work with a reduced dataset. This aims at decreasing the computation time of Monte Carlo simulations. To keep a sufficient number of defects to be inspected, 7 particular parts were chosen. From an initial total number of 77 defects spread between the 20 parts, the remaining set of 7 parts contains 33 of those 77 defects.

This smaller dataset allows to reduce the computation time to 4 minutes while maintaining a sufficient number of defects to run Monte Carlo simulations properly. The distinction between different defects such as galling, scoring, nicks, etc. is not a part of the scope. The distribution of the number of defects for the different parts and acquisitions is presented in table 4.3. These defects were initially inspected and labelled by operators of Rolls-Royce Canada Limited. As it can be seen on the schematic of the fan blade, the acquisitions overlap leading to the same defect detected multiple times (figure 4.10). It explains the difference between the total number of defects per part and the real one in table 4.3.

Table 4.3 Distribution of the number of defects for each part and each acquisition

Acquisition	Part 1	Part 2	Part 3	Part 4	Part 5	Part 6	Part 7	Total
#1	1	1	2	2	0	1	1	8
#2	2	2	1	3	1	1	2	12
#3	2	2	0	0	2	2	0	8
#4	1	0	1	1	0	0	1	4
#5	0	0	1	0	0	0	0	1
#6	2	1	1	0	3	0	0	7
#7	0	1	4	3	0	0	2	10
#8	0	0	5	3	0	1	3	12
Total	8	7	15	12	6	5	9	62
Real number of defects	5	3	7	5	4	5	4	33

Since the entire detection routine of a fan blade involves a large number of algorithms, the assessment of all the uncertainties is not carried out due to the time and means constraints specific to the realization of a master's project. Hence, AV&R asked us to focus the analysis on one sensitive algorithm called *Slope Detection*. It is a deterministic edge detection algorithm. This algorithm is studied as a black box, thus we do not have access to it. To detect the surface defects, photometry is used. It consists in illuminating the fan blade from different directions while acquiring images to combine them into one to highlight those defects due to a difference in the reflection properties [66]. Therefore, a ring of 8 LED encircles the camera to allow different lighting settings. For each acquisition, 8 images are taken with one different LED on and are then combined to be used as input image of the *Slope Detection* algorithm. Each of those 8 images is called a *Grab*.

Among all the parameters of *Slope Detection*, three particular factors seem to be impacting in the detection of defects according to AV&R's inspection team. Their values are different from one acquisition to another and they are slightly modified between versions when it gives better results. These key parameters and their working range are written in table 4.4. The filter size represents the kernel size of a filter used. The minimum defect size is the smallest size of blob to consider as a defect. The threshold is the value above which the pixel intensity is associated to one, zero otherwise.

Table 4.4 Type and working range of impacting parameters of the *Slope Detection* algorithm

Parameter	Type	Working range
Threshold	Integer	500-4500
Filter size	Odd integer	1-21
Minimum defect size	Integer	0-500

The outputs of the algorithm are the number of detected defects and the number of false detections. False detections can be caused by any artifact in images. The aim of the detection algorithm is to minimize those while maximizing the detection rate. AV&R's software returns the total number of detected defects and the total number of false detections for all the parts but those outputs can be extracted for each acquisition and for each part. Another parameter to take into account is the contour quality. For each detected defect, a certain value for this contour is associated. This parameter is an area ratio between the contour surrounding the defect and the defect itself as illustrated in figure 4.11. On the left hand side of the figure the contour quality variable is equal to one since the defect is perfectly encircled while on the right hand side the same ratio is equal to 3. Ideally, this parameter should be as close to one as possible. The latest version of the slope detection algorithm has a narrower range to avoid the use of the contour quality but was not used in this project.

The table 4.5 presents the detection baseline used throughout this project when the default values of the three key parameters are used and no input uncertainty is added to the images. Considering the 33 defects distributed on the 7 parts, the detection rate is 45.45%. The average number of false detections per blade is 17.29.

Another detection algorithm called *Surface defect* can be added to the previous algorithm. This deterministic algorithm aims at distinguishing defects by comparing them with non damaged surfaces of several parts. The three key parameters are thus applied on the image difference between the damaged image and the clean image.

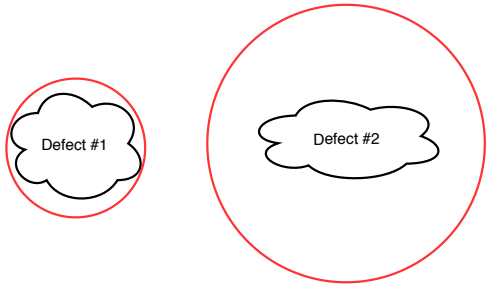


Figure 4.11 Contour quality representation

Table 4.5 Detection baseline for the outputs of the *Slope Detection* algorithm with no uncertainty added to input images

Output	Total on the 7 blades
Detected defects	15
Number of false detections	121

4.2.1 Sensitivity of *Slope Detection* outputs to the three key parameters

To understand the intrinsic principle of this detection algorithm, we need to know how sensitive are the number of detected defects and the number of false detections to a variation of the three mentioned key parameters. Since this algorithm is a black box, one cannot extract the model. We can only obtain detection results for a specified set of parameters. Furthermore, the contour quality variable needs to be taken into account in the analysis to avoid trivial results. Indeed, a detection rate of 100% and a few false detections can be obtained but the contour quality will be too large. In fact, the contour quality variable will be as large as the size of the image, hence those pseudo-optimal detection results.

Each of the 8 acquisitions have different values for the threshold, the minimum defect size and the filter kernel size. A slight variation of those parameters has thus a different impact on the results. We decide to focus on the first acquisition since the following procedure is applicable to the other acquisitions. First, we want to characterize quantitatively the aforementioned sensitivity. Due to the vast working range of each parameter, a randomized search could be tiresome and useless. Therefore, two methods are used to compute the impact factor of each key parameter on the detection outputs : Spearman's correlation and feature importance.

The first step is to generate a dataset of random combinations to cover as much as possible the entire working range. The computation time of one iteration of the *Slope Detection* algorithm is about 4 minutes. Moreover, for each iteration the values of the key parameters are set by hand. Due to those two time constraints, only 450 random combinations of the key parameters are generated. Considering an average time of 10 seconds to change values between each iteration, the total time to generate those 450 combinations is about 32 hours. The model is run and one obtains the associated detected defects, the false detections and the contour quality variable. The three key parameters are the features or the independent variables X_i of our model while the number of detected defects, the number of false detections and the contour quality are the dependent variables Y_i.

Correlation

Since the data do not follow a normal distribution and presents a lot of extreme values, Pearson's correlation cannot be used [103]. Hence, Spearman's correlation seems to be more appropriate according to [103]. The Spearman's correlation coefficient is defined as following (equation 4.24)

$$r = 1 - \frac{\sum_{i=1}^{n}(rank(X_i) - rank(Y_i))^2}{n(n^2-1)} \tag{4.24}$$

With n being the number of observations, X_i the inputs (the key parameters) and Y_i the detection outputs. The correlation coefficient varies from -1 to +1. An absolute value close to one means that both variables are highly correlated while not correlated if the coefficient is close to 0. A Spearman's correlation coefficient close to 0.5 in absolute value is described as a moderate correlation in [103].

Feature importance

To evaluate the contribution of each key parameter to the detection outputs, we implement a random forest algorithm to compute the feature importance. A random forest is a machine learning ensemble algorithm used either for classification or regression as described in [104] and applied for failure detection in [105]. In our case, we want to build a model that fits the data hence regression is used. A random forest is composed of numerous binary tree predictors, called decision trees [106]. Each decision tree is trained on random samples of the dataset through what is called bootstrap aggregating and returns a predicted value. The total prediction of the random forest is then based on the average of each decision tree prediction. To assess the importance score of a variable, out-of-bags samples of this variable are permuted and the associated mean square error is computed [104]. Out-of-bags samples

of a decision tree are the samples not used during the step of bootstrap aggregating. The mean square error is the mean squared difference between the real value of an output and its prediction.

The random forest method is implemented on Python using the *Scikit-learn* library. The dataset is divided into a training and a test set for the dependent and independent variables. A conventional 80%-20% split is chosen. Afterward, a scaling of the features is performed due to the differences in orders of magnitude between them. Finally, the random forest regressor is instantiated with 1000 decision trees and the variable importance is computed. The hyperparameters were not optimized, i.e. the parameters of the random forest regressor.

4.3 Uncertainty modeling of image pixels

The acquisition of images with CCD or CMOS cameras presents different sources of uncertainty. As mentioned in the GUM, each process contains random and systematic effects [13]. These latter can more easily be corrected compared to this random variability. As mentioned in [107, 108], random and systematic effects cause a dispersion of intensity for each pixel, i.e a standard uncertainty. The acquired image $I_{ACQ}(i,j)$ is a matrix of pixels differing from the expected image $I_0(i,j)$ due to two systematic effects: the response matrix $r(i,j)$ and the dark frame $I_D(i,j)$ (equation 4.25). These two effects rely on the lens and the sensor used.

$$I_{ACQ}(i,j) = I_D(i,j) + r(i,j)I_0(i,j) \tag{4.25}$$

The dark frame corresponds to the matrix obtained when there is no illumination on the observed scene. Since photons do not hit the CCD sensor in a uniform way, the response matrix takes that variability into account. These two matrices can be evaluated experimentally. Since the implementation of a new routine takes some time to put in place, one assumes that these two systematic effects have been corrected. Hence, the obtained images is the expected one $I_0(i,j)$.

To compute the uncertainty of the acquired images, a type A evaluation is carried out. For that purpose, the acquisition process of the critical area of the fan blade is repeated several times. To obtain a sufficient number of iterations while keeping a reasonable cycle time, a hundred images per zone is taken. To take the environmental effects into account when evaluating the uncertainty of the images, a random delay up to 10 minutes is set between each image capture. For each of the 8 acquisitions composing the critical zone and for each

Grab, a matrix of the average value of each pixel is obtained in equation 4.26.

$$\overline{I_0}(i,j) = \frac{1}{100}\sum_{k=1}^{100} I_{0,k}(i,j) \tag{4.26}$$

For each pixel, a distribution is thus associated from which the standard uncertainty is the standard deviation (equation 4.27)

$$u_{I_0}(i,j) = \frac{1}{100}\sqrt{\frac{1}{100-1}\sum_{k=1}^{100}(I_{0,k}(i,j) - \overline{I_{0,k}}(i,j))^2} \tag{4.27}$$

Thus, 64 pixel uncertainty matrices are computed.

4.4 Uncertainty evaluation using Monte Carlo method

To evaluate the uncertainty of calibration methods and the detection routine, Monte Carlo simulations are carried out by propagating distributions. The assessment of those uncertainties cannot be carried out using the law of uncertainty propagation for three reasons: some models are not available such as the detection algorithms, there are several complex algorithms involved in the inspection routine, there are sources of non linearity. The working principle of the Monte Carlo method is detailed in section 2. Regarding the calibration methods, the outputs uncertainty of the focus algorithm, Zhang's method and the POSIT algorithm is evaluated. Then, the uncertainty evaluation of the detection routine is presented.

4.4.1 Calibration methods

Since the calibration methods are implemented in MATLAB, Monte Carlo simulations are added to the existing codes. For the calibration of the focus and the depth of field, the black vertical lines are the inputs of the model. The intensity uncertainty of images taken by the camera not being calibrated is not known. We thus assume an independent normal distribution for each pixel with a consistent standard deviation. The variation of the focus and the depth of field uncertainties for input pixels uncertainty up to 15 is analyzed.

As mentioned in [80], the number of trials must be 10,000 bigger than $\frac{1}{1-p}$, with p being the chosen coverage probability. Since, the coverage probability is set to 95%, the minimum number of trials is 200,000. Thus, 225,000 Monte Carlo simulations are run to obtain the expanded uncertainty of the depth of field and the focus position. Furthermore, the impact of an increasing standard deviation of the Gaussian noise added is evaluated and visualized.

In Zhang's method [38], the inputs are the 3D control points and the associated image points. We decide to add a Gaussian noise with zero mean and a standard deviation of respectively 1 pixel and 1 mm. According to [82], the image noise encountered in applications are rarely larger than 0.5 pixel. The worst-case scenario is thus considered by choosing a standard deviation of 1 pixel. Furthermore, the reprojection error obtained is around 0.28 pixel which is smaller than the noise added. For the 3D control points, the squares sides are measured with a caliper. Thus, a standard deviation of 1 mm is reasonable. Due to the computation time of this algorithm, 80,000 Monte Carlo simulations were run to obtain the expanded uncertainty of the focal length and the image center.

For the POSIT algorithm, the same assumptions about the input uncertainties of the control points images are used. For the 3D points, the resolution of the 3D printer MakerGear M2 is up to 0.25 mm for the layers according to their website. Nevertheless, they mention that it is not the only parameter to take into account when considering the overall resolution of a print. Therefore, a Gaussian noise with zero mean and a standard deviation of 0.5 mm was added to consider all the possible effects. The uncertainties obtained for the evaluation of Zhang's method are used as input uncertainties for the focal length and the image center. The number of trials chosen is 300,000 to have enough data. To evaluate the impact of each input parameter, a sensitivity analysis is also carried out for the 4 parameters.

4.4.2 Detection routine

The number of detected defects and the number of false detections of the *Slope Detection* algorithm are the variables of interest for which we want to compute the uncertainty. In this case, Monte Carlo simulations are directly implemented into the software of AV&R since the algorithm was not adapted to another programming language. The inputs of this algorithm are the images of the 8 acquisitions obtained from the photometry. The associated uncertainties are thus the ones computed in section 4.3. The working principle of this uncertainty evaluation for one acquisition on the critical zone of the blade is depicted in figure 4.12.

The standard uncertainty of each pixel of the input image is denoted $u(x_i)$ while the uncertainty of the number of detected defects and the number of false detections are $u(y_i)$. The values of the three key parameters remain fixed for all the simulations and correspond to the default values implemented when the version was created. Furthermore, the impact of the surface finish on the detection uncertainties is not taken into account first but is analyzed afterward.

Two versions of the input uncertainty implementation were developed hence two versions of the Monte Carlo simulations. In both versions, the standard uncertainty of each pixel is

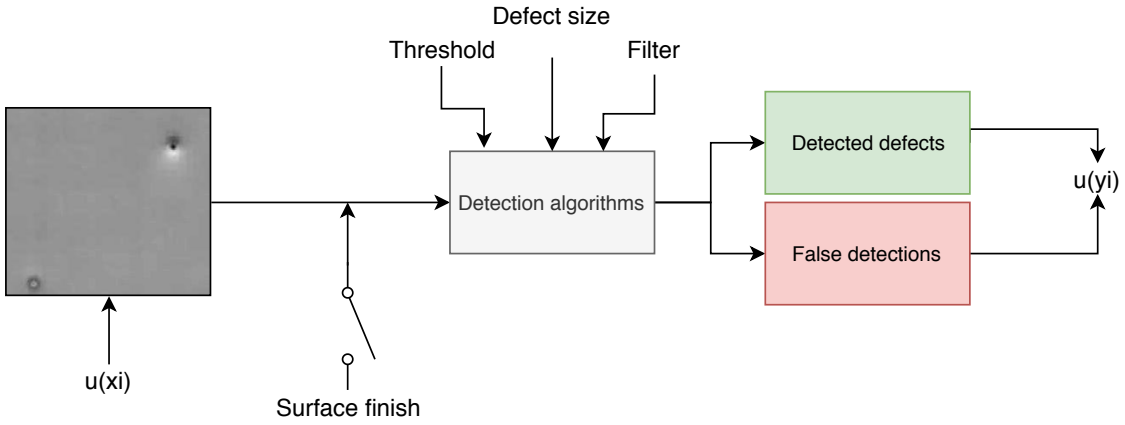


Figure 4.12 Working principle of the detection routine uncertainty evaluation for one acquisition

added to the images via a text file. Since the images resolution is 2,048 x 2,048, each text file contains a table with 2,048 rows and 2,048 columns that we call an uncertainty matrix. Each cell contains a sample of the corresponding pixel distribution. Hence, for each iteration a text file is created with a different sample of the each pixel distribution. The particular characteristic of the first version is that, for one iteration, the same uncertainty matrix (contained in the text file) is applied for all 8 acquisitions. In practice, the computed uncertainty matrix for the pixels vary from one acquisition to another and from one *Grab* to another. Indeed, we observed that the resulting detection uncertainties are different depending on which input uncertainty matrix was added. The second version takes thus into account that the corresponding uncertainty matrix is added to each *Grab* of each acquisition.

The size of one text file is around 26 Mb which limits the number of Monte Carlo simulations due to limited storage. Indeed, considering the minimum number of simulations of 200,000, around 5 Tb needs to be allocated for that task for each acquisition *Grab*. Furthermore, the computation time of one iteration for 8 acquisitions on 7 parts lasts 4 minutes. The number of simulations is then limited and was chosen to be 1,000 even though it does not meet the requirements of GUM [80]. Furthermore, we reduce the number of simulations to 100 and compare the obtained output uncertainties. The sensitivity of the detection uncertainties to each acquisition is studied by adding the uncertainty matrix to one acquisition at a time and compute the corresponding uncertainty. Due to the required computation time, only 100 simulations are done.

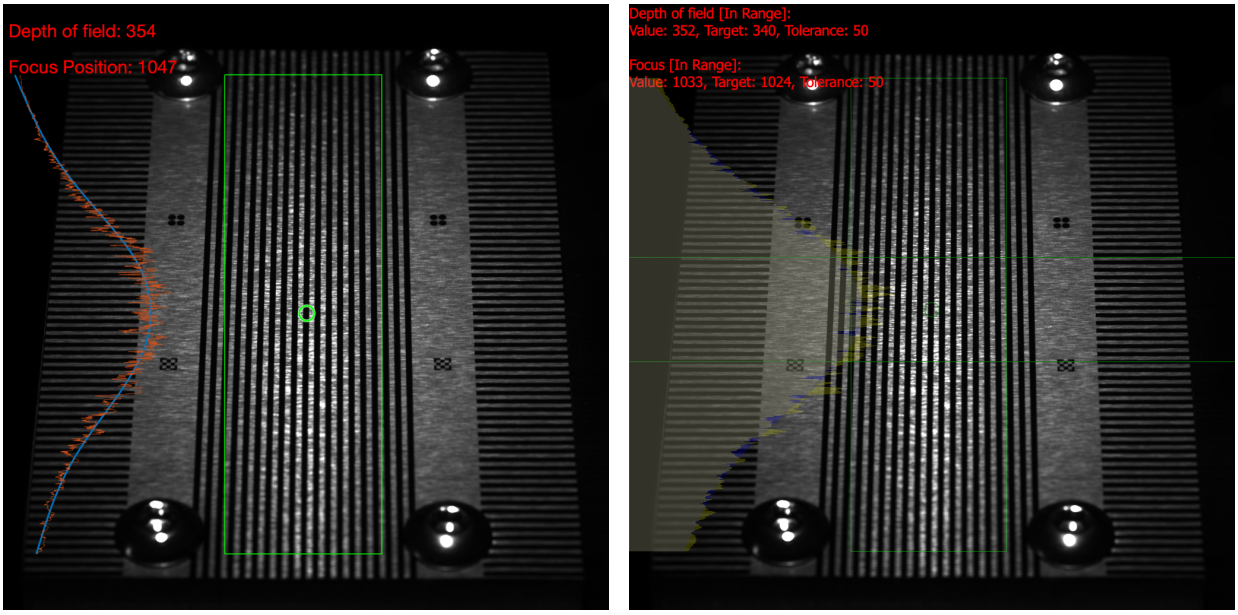
Different scenarios are then analyzed to get a better insight of those detection uncertainties. First, the second algorithm *Surface Defect* is added to the routine and the Monte Carlo simulations are run again to evaluate the impact of the *Slope Detection* algorithm. To conclude on the impact of the blade surface finish on the detection uncertainties, a set of two unrepaired fan blades is used to compute new detection uncertainties and compared it to the ones obtained for the original blade set, i.e. the repaired ones.

CHAPTER 5 RESULTS

5.1 Focus and depth of field implementation

5.1.1 Comparison of the output results

The focus position and the depth of field obtained with our implementation in MATLAB are close to the results obtained by AV&R (figure 5.1). The green rectangle in the figure represents the region of interest. Since the focus position is a pixel on the image, a circle is drawn with this pixel as center. Even though the DOF and the focus are not exactly the same, they are in the tolerance zone. Moreover, the relative error for the focus and the DOF are respectively 1.35% and 0.57%. The differences in those values might come from the way the spline fitting algorithm is implemented in MATLAB compared to AV&R's software.



(a) (b)

Figure 5.1 Comparison of results obtained for the DOF and the focus position: a) MATLAB implementation b) AV&R's implementation © 2019 AV&R

The table 5.1 summarizes the statistics of the implemented spline fitting and shows that the model fits correctly to the data. To investigate the difference in the results, it might be interesting to compare those statistics with the ones obtained by AV&R.

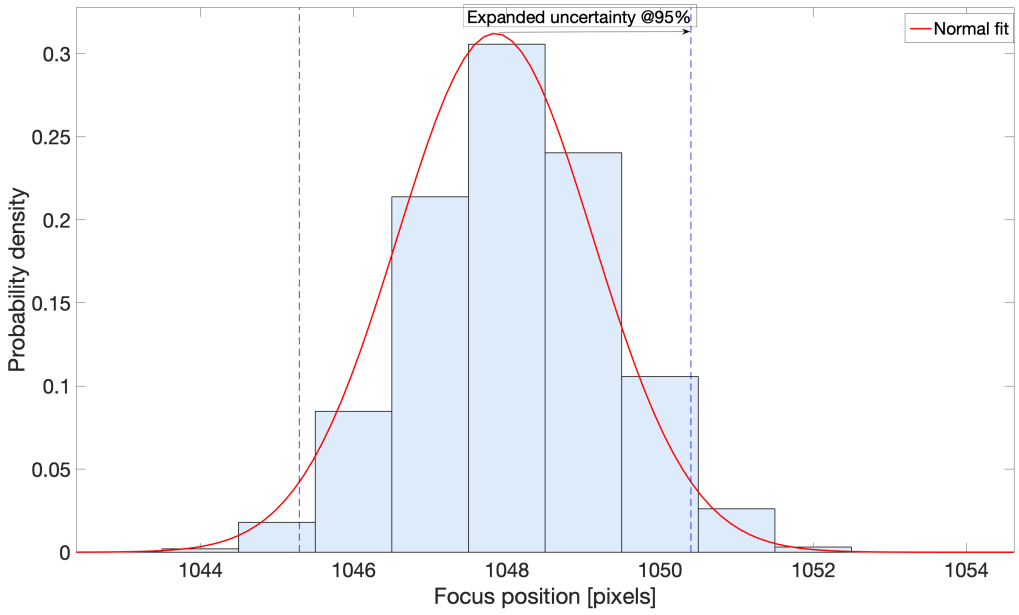
Table 5.1 Statistics of the spline fitting

Parameter	Value
R^2	0.97
Adjusted R^2	0.97
RMSE	10.09

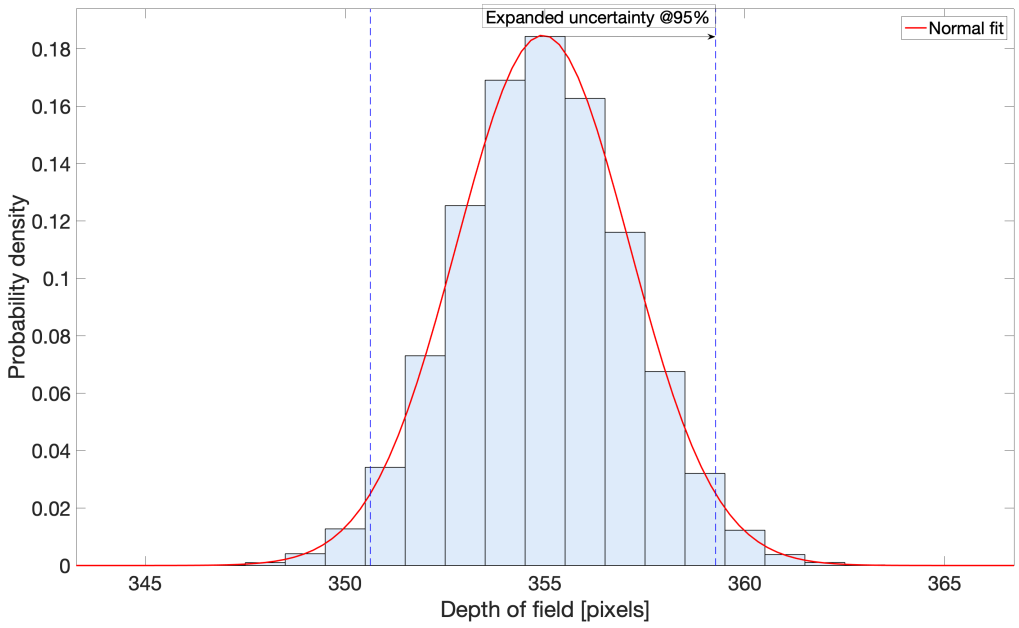
5.1.2 Uncertainty evaluation

Figure 5.2 depicts examples of probability distributions obtained for the focus position and the DOF when the added Gaussian noise has a standard deviation of 1. To obtain a coverage probability of 95% while running Monte Carlo simulations, the minimum number of trials needs to be set to 200,000. Hence, 225,000 trials were used to plot both output parameters. These distributions can be normally fitted. With a coverage probability of 95%, the expanded uncertainty of the focus position and the depth of field are respectively 4.32 pixels and 2.56 pixels. Since this large number of simulations lasts 45 hours, it was decided to reduce it to perform the other required simulations. Furthermore, the expanded uncertainties have a relative error of half a percent when choosing 10,000 trials compared to the 225,000 trials.

The figures 5.3 and 5.4 show the variation of the mean and the expanded uncertainty of respectively the focus position and the depth of field for different standard deviations of the Gaussian noise added. The red line is the target set by AV&R while the yellow area being the tolerance zone. We can see that the uncertainty of both outputs grows with the increase in the noise standard deviation. An input standard uncertainty around 7 is the acceptable limit to have both outputs in their tolerance zone. The input uncertainty of the images is unknown at this stage of the project. AV&R needs to make sure that this uncertainty does not exceed the limit at the risk of having to modify their algorithm. Furthermore, one can observe that the mean value of the focus and the DOF get larger with the noise increase. It means that the focus point is lowering on the image while the acceptable focused zone increases.



(a)



(b)

Figure 5.2 Probability density function of the focus position (b) and the DOF (b) obtained for 225,000 trials

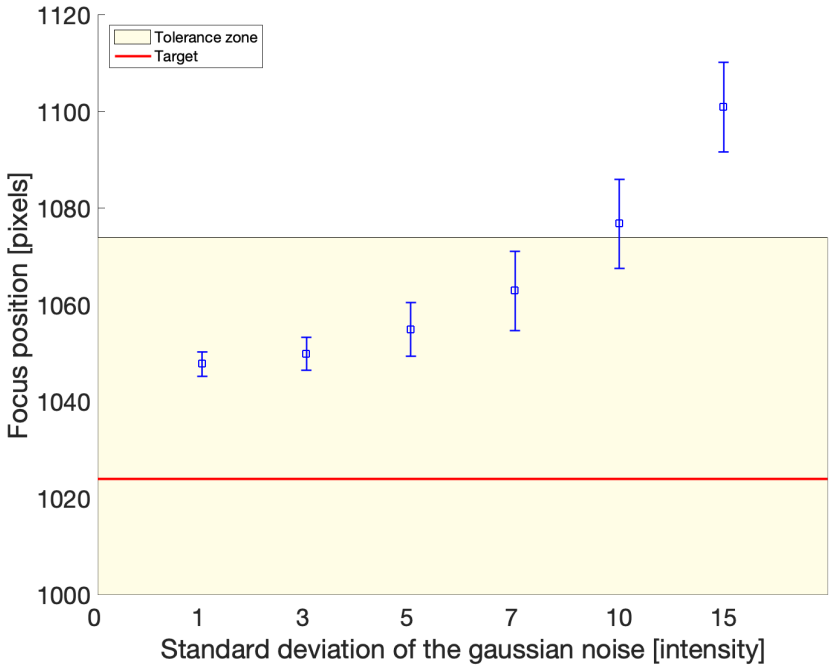


Figure 5.3 Variation of the mean and the expanded uncertainty of the focus for an increasing Gaussian noise standard deviation

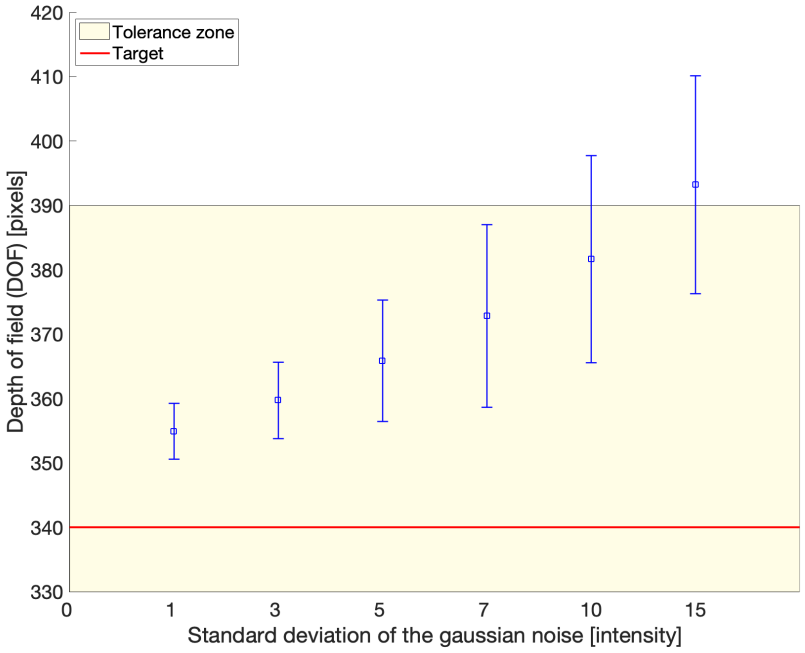


Figure 5.4 Variation of the mean and the expanded uncertainty of the DOF for an increasing Gaussian noise standard deviation

5.2 Camera parameters obtained with implemented methods

5.2.1 Intrinsic parameters (Zhang's method)

The intrinsic parameters of the purchased camera computed with Zhang's method are shown in equation 5.1. The skew coefficient is close to zero which means that the CCD axes are almost perpendicular. Indeed, the angle between them is 89.99°. The other parameters are expressed in pixels. The effective focal length of 1201.38 pixels is used for the POSIT algorithm.

$$\mathbf{K} = \begin{pmatrix} c_u f & \alpha & u_c \\ 0 & c_v f & v_c \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1201.38 & 0.18 & 622.91 \\ 0 & 1197.88 & 369.95 \\ 0 & 0 & 1 \end{pmatrix} \tag{5.1}$$

The radial and tangential distortion coefficients are shown in equation 5.2. As expected, the distortion coefficients are low.

$$\begin{cases} [\kappa_1 \ \kappa_2] = [0.05 \ \ -0.04] \\ [p_1 \ p_2] = [-0.01 \ \ -0.01] \end{cases} \tag{5.2}$$

The figure 5.5 shows the individual reprojection error for each of the acquired images used for the calibration and the mean reprojection error of 0.28 pixels. None of the errors exceed one pixel which confirms the order of magnitude of the noise added for Monte Carlo simulations.

Figure 5.5 Mean reprojection error of the acquired images of Zhang's method

5.2.2 Extrinsic parameters (POSIT)

Based on the Appendix B, we can see that the *World* reference frame is defined in the same configuration as the camera reference frame. Since the camera is placed right in front of the target, the rotation matrix should ideally be the identity matrix, i.e. Euler angles of 0°. Furthermore, we placed the camera 22 cm from the target along the z axis. Therefore, the optical center of the camera should be around 21 cm. We measured the distance from the origin of the *World* reference frame the along the x axis and the y axis: 29.5 mm and 5.45 mm. This estimated pose, called the targeted pose, is roughly measured and corresponds to the case where the camera is perfectly aligned. The table 5.2 presents the position and the orientation of the *World* reference frame with respect to the camera reference frame obtained with the POSIT algorithm. The results are compared with the targeted pose.

Table 5.2 Estimated pose of the camera returned by the POSIT algorithm

Method	Euler angles ZYX [°]	Translation vector **T** [mm]
Targeted pose	[0 0 0]	[-29.50 -5.45 210.00]
POSIT	[0.33 -2.65 2.85]	[-25.79 -4.74 207.29]

The average Euclidean distance E computed for this pose is 0.60 pixels. One can assess this value by comparing the image points of the control points and the corresponding projected points obtained (table 5.3). Projected image points are close to the actual ones.

Table 5.3 Comparison of the projected points obtained with actual image points

Control point	Actual image points (u_i, v_i)	Projected image points (u_i^p, v_i^p)
#1	(-149.43 -27.45)	(-149.43 -27.45)
#2	(-16.28 -26.98)	(-16.24 -26.54)
#3	(-149.03 104.63)	(-149.68 105.06)
#4	(-17.54 106.41)	(-17.21 105.29)

5.2.3 Uncertainty evaluation

The table 5.4 compares the expanded uncertainty at 95% of the focal length and the principal point obtained with the Calibrator App and using the Monte Carlo simulations. There is a difference of one order of magnitude between the obtained uncertainties. The results obtained with Monte Carlo simulations are used to evaluate the uncertainty of the POSIT algorithm since the Calibrator App does not specify how it is computed and what is the input uncertainty used.

Table 5.4 Expanded uncertainty (@95%) comparison of Zhang's method

Method	Expanded uncertainty of focal length (@95%) [pixels]	Expanded uncertainty of the principal point (@95%) [pixels]
Camera Calibrator App	1.75	(1.65,1.90)
Monte Carlo	13.11	(12.36,14.55)

The table 5.5 shows the expanded uncertainty of the camera pose obtained with 300,000 Monte Carlo simulations of the POSIT algorithm. We can see that the expanded uncertainty of Euler angles Θ and Φ are excessive since it is around 23°. Moreover, t_z presents an uncertainty 2.3 times bigger than the ones obtained for t_x and t_y. The mean value of the error E for those 300,000 simulations is 1.72 pixels and the maximum one is 7.74 pixels. Therefore, these high numbers are not due to a non-convergence of the algorithm.

Table 5.5 Expanded uncertainty (@95%) for the estimated camera pose

Parameter	Euler angles ZYX [°]			Translation vector **T** [mm]		
	Ψ	Θ	Φ	t_x	t_y	t_z
	2.82	22.47	23.33	4.37	4.95	11.42

To understand the reason behind these high values, a sensitivity analysis of the POSIT algorithm is done according to [80]. The sensitivity coefficients of each parameter are shown in table 5.6. The two Euler angles Θ and Φ are mainly sensitive to the noise added to the 3D control points but also to the noise added to the corresponding image points. The z coordinate of the translation vector t_z is sensitive to the noise added to the 3D and 2D points but also to the focal length noise found with Zhang's method.

Table 5.6 Sensitivity analysis of the POSIT algorithm

Parameter	Euler angles ZYX			Translation vector **T**		
	Ψ	Θ	Φ	t_x	t_y	t_z
Focal length f	0.00	0.01	0.00	0.00	0.00	0.17
Center coordinate u_c	0.00	0.05	0.08	0.17	0	0.01
Center coordinate v_c	0.00	0.01	0.02	0.00	0.17	0.01
2D points	0.45	5.81	5.84	0.21	0.17	2.01
3D points	2.65	21.83	22.55	1.25	0.23	10.55

5.3 Uncertainty evaluation of image pixels

For each *Grab* of each acquisition, 100 images were acquired over a night with a variable capture time to compute the uncertainty of all the pixels. Those are 16-bit images and there are therefore 65,536 intensity levels. The figure 5.6 depicts the intensity distribution of one random pixel of one acquisition *Grab* acquired 100 times. The mean value is 4,746.24 and the standard uncertainty is 172.11. The pixels across the image have different distributions and there are not enough samples to conclude on the type of continuous probability distribution observed.

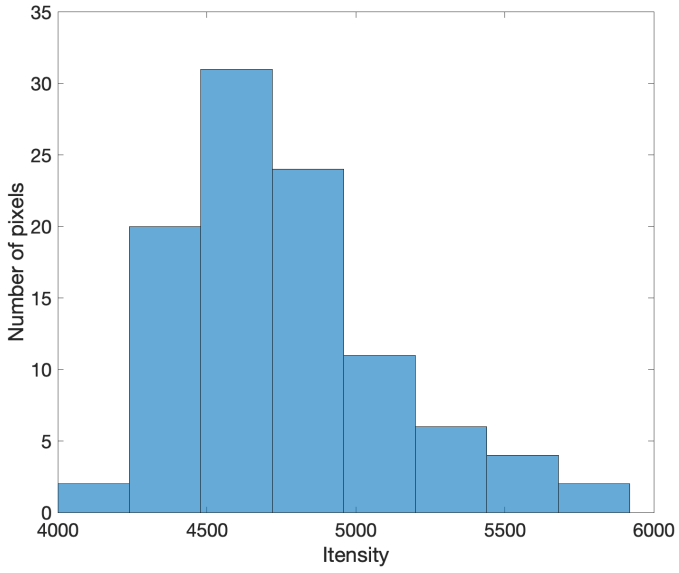


Figure 5.6 Distribution of one pixel intensity for 100 acquired images of the same zone

The figure 5.7 shows the pixels standard uncertainty of a part of the first *Grab* acquired for the fifth acquisition. For the sake of readability only a 300x300 square of the original 2,048x2,048 image is shown. It can nevertheless be observed that there is a disparity of the pixels standard uncertainty regarding the zone of the image especially when looking at the drop on the right bottom corner of the image. This conclusion can be drawn for all the *Grabs* of all the acquisitions. Thus, for each *Grab* of each acquisition we obtain a different 2,048x2,048 uncertainty matrix that is used as input uncertainties of the detection routine. This uncertainty disparity might be caused by the repeatability error of the robot. Between each acquisition, the robot might not return to the exact same location leading to high variations in pixels intensity.

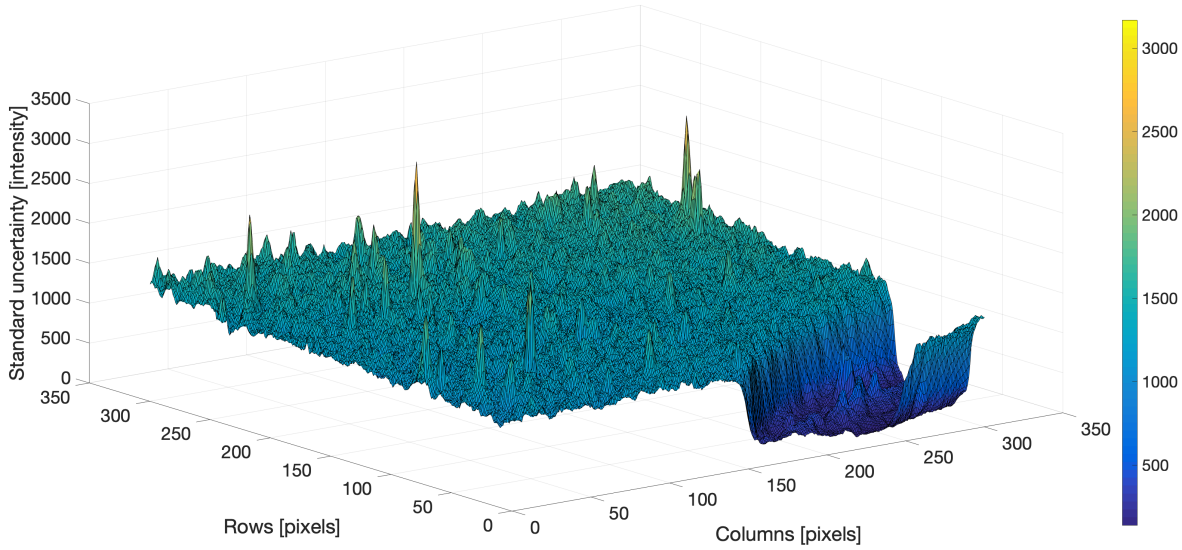


Figure 5.7 Pixels standard uncertainty of the first *Grab* of the fifth acquisition

5.4 Sensitivity of *Slope Detection* results

The table 5.7 presents the results of Spearman's correlation and the feature importance. Even though the units of the results are not the same, the conclusions drawn are similar. The feature importance method implemented using the random forest regressor states that there is no key parameter variation that predominates by far the detection outputs. Nevertheless, the number of detected defects and the number of false detections are more sensitive to a filter or a threshold variation than a change in the minimum defect size. The correlation

coefficients are the average of the correlation coefficients obtained for the three detection outputs. We observe that the outputs present a negative moderate correlation with all the three key parameters. The correlation strength follows the same conclusion as the feature importance results.

Table 5.7 Sensitivity of the detection outputs to the three key parameters of *Slope Detection*

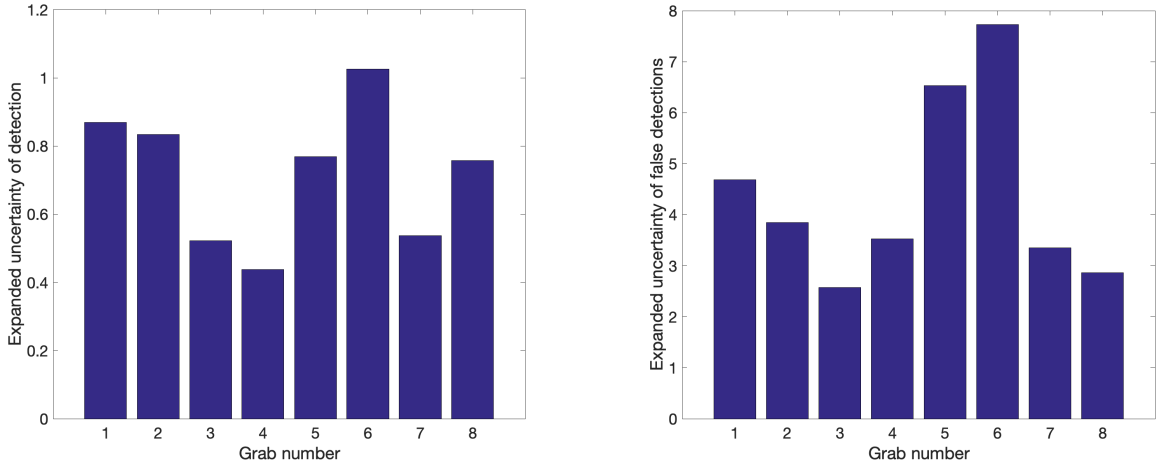
Method	Filter	Threshold	Minimum defect size
Feature importance	36.40%	35.80%	27.80%
Correlation	-0.52	-0.45	-0.40

5.5 Uncertainty evaluation of the detection routine

The figures 5.8 and 5.9 show detection uncertainties obtained for the first implemented version of Monte Carlo simulations for which the same uncertainty matrix file is applied for all *Grabs* of all acquisitions. For both figures, the expanded uncertainty with a coverage probability of 95% is shown. The figure 5.8 depicts the variation of the number of detected defects and the number of false detections uncertainties when we add the uncertainty matrix of one particular *Grab* to all of them for the acquisition number 3. We can notice a difference in the detection uncertainties depending on which *Grab* uncertainty matrix is used. This variation in the results shows that the lighting direction has a significant impact on the detection uncertainties. Indeed, for that acquisition the order of magnitude of the number of detected defects can change depending on the uncertainty matrix used which could cause an underestimate of the results.

The figure 5.9 presents the uncertainty variation of the same detection outputs but for the same *Grab* of different acquisitions. Therefore, it results that for the same lighting conditions, the detection uncertainties are not constant over the most critical zone of the fan blade. These two figures led to the implementation of the new version of Monte Carlo simulations for which each *Grab* of each acquisition has its corresponding uncertainty matrix as added noise.

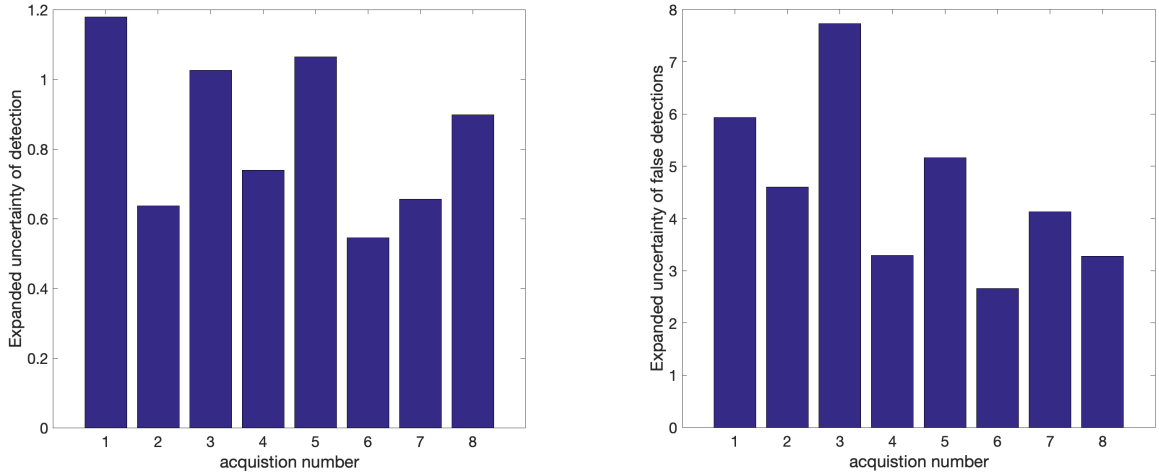
The table 5.8 presents the expanded uncertainty of the number of detected defects and the number of false detections for the second version of Monte Carlo simulations. Unlike the first version for which only the *Slope Detection* algorithm was studied, we also compare the detection uncertainties when adding the *Surface Defect* algorithm to the detection routine. If



(a) Number of detected defects

(b) Number of false detections

Figure 5.8 Expanded uncertainty variation for the different *Grabs* of the acquisition number 3



(a) Number of detected defects

(b) Number of false detections

Figure 5.9 Expanded uncertainty variation for the *Grab* number 6 of all the acquisitions

one refers to the table 4.5, the detection uncertainties are not significantly high compared to the baseline results. Indeed, considering the total number of false detections, the associated uncertainty does not even represent one false detection per blade. The effect is slightly bigger for the total number of detected defects since it could affect it by about 4%. The uncertainty of the false detections is more sensitive to the addition of the *Surface Defect* algorithm than

the number of detected defects.

For the uncertainty evaluation of the *Slope detection* algorithm, the difference observed between 100 and 1,000 iterations is small. Indeed, the relative error is 1.47% for the number of detected defects and 5.5 % for the number of false detections. When the second algorithm is added to the routine, one observes an increase in the uncertainties but the relative errors are still small and respectively 2.07% and -3.97%.

Table 5.8 Comparison of the expanded uncertainty of the detection outputs for different numbers of iterations and with the addition of the *Surface Defect* algorithm

Algorithm	Number of iterations	Expanded uncertainty of the number of detected defects (@95%)	Expanded uncertainty of the false detections (@95%)
Slope Detection	100	1.38	5.93
Slope Detection	1000	1.37	5.62
Slope Detection + Surface Defect	100	1.69	8.34
Slope Detection + Surface Defect	1000	1.66	8.67

5.5.1 Effect of the defect location

The table 5.9 presents the expanded uncertainty distribution of the number of detected defects and the number of false detections according to the acquired acquisitions for the *Slope Detection* algorithm. The acquisition numbers 4 and 5 do not present any detection uncertainty for both outputs. By looking at the table 4.3, we can notice that these acquisitions correspond to the ones with the fewest number of defects.

Nevertheless, the acquisition number 8 presents no uncertainty for the detection although there are 12 defects among the 7 parts for this acquisition. This can be explained by the process used to distinguish the detected defects for each acquisition. As mentioned in table 4.3, acquisitions overlap on the fan blade critical area. The defects of the acquisition number 8 are observed on the surroundings acquisitions. The process of removing duplicates causes thus the observation of zero defects, rarely one. This leads to the presented uncertainties. The zone composed of the acquisitions number 1, 2 and 6 has a higher concentration of detection uncertainties. The further away from this zone, the greater the detection uncertainties tend

Table 5.9 Distribution of the expanded uncertainty of the detection outputs for the different acquisitions (*Slope Detection*)

Acquisition number	Expanded uncertainty of the number of detected defects (@95%)	Expanded uncertainty of the false detections (@95%)
1	1.02	3.53
2	0.99	2.38
3	0.75	1.24
4	0.00	0.00
5	0.00	0.00
6	1.09	2.54
7	0.27	1.73
8	0.00	1.74

to decrease.

5.5.2 Effect of the surface finish

The table 5.10 shows the expanded uncertainty of the number of detected defects and the number of false detections for 1,000 iterations of the *Slope Detection* algorithm when a set of two fan blades with a rough surface finish is used, i.e. unrepaired blades. These detection uncertainties are significantly lower than the ones obtained for the original set of blades which presents a different surface finish since they were repaired (table 5.8). The repair seems to have a negative impact on the detection uncertainties. Indeed, the repair process of the fan blade might add some artifacts to the acquired images resulting in this uncertainty increase. Nevertheless, the unrepaired dataset is not comparable to the repaired one to assert this observation for a larger number of parts.

Table 5.10 Expanded uncertainty of the *Slope Detection* algorithm for two unrepaired fan blades

Algorithm	Expanded uncertainty of the number of detected defects (@95%)	Expanded uncertainty of the false detections (@95%)
Slope Detection	0.27	2.42

5.5.3 Sensitivity analysis

The figures 5.10 and 5.11 depict the sensitivity analysis of the detection uncertainties for each acquisition. Thus, the uncertainty matrix is added to each acquisition individually to compute the outputs uncertainties. For the number of detected defects, we observe a constant expanded uncertainty except for two acquisitions (Figure 5.10). The acquisition number 1 presents the highest impact on the total uncertainty while the acquisition number 3 the lowest. However, the expanded uncertainty of the false detections is relatively close from one acquisition to another (Figure 5.11).

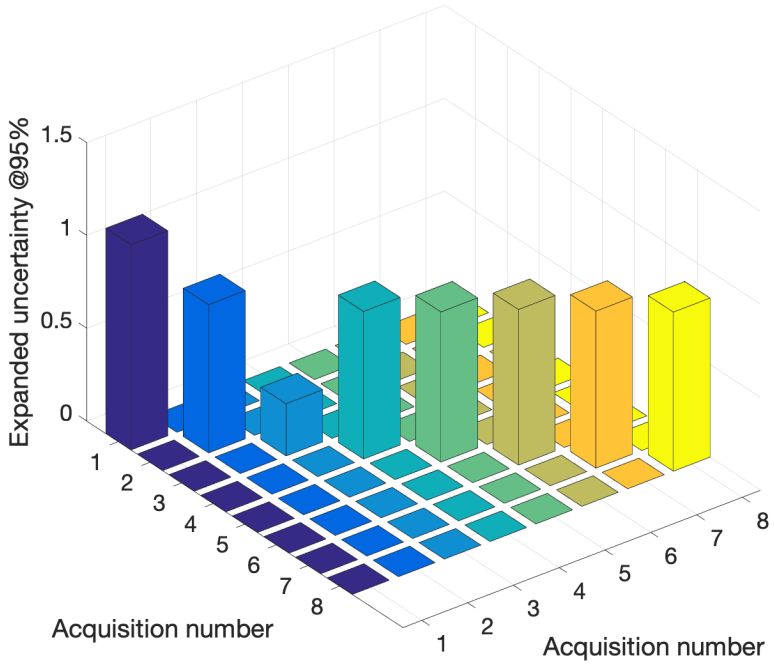


Figure 5.10 Sensitivity analysis of the detected defects expanded uncertainty (@95%)

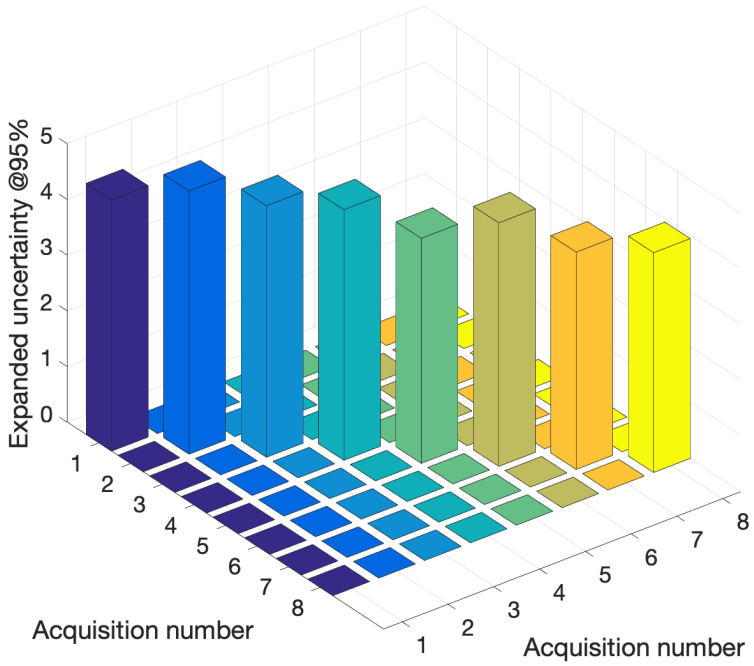


Figure 5.11 Sensitivity analysis of the false detections expanded uncertainty (@95%)

CHAPTER 6 DISCUSSION

The objective of this project was to quantify the uncertainty of the focus and depth of field calibration (S.O.1) and the detection outputs of the *Slope Detection* algorithm; for defects detected on the most critical area of a fan blade. Moreover, this project also aimed at developing a camera pose estimation method and quantifying the associated uncertainty to overcome the lack of measurements required to completely model the extrinsic camera calibration developed by AV&R (S.O.2). Both objectives were achieved. The last specific objective (S.O.3) consisting in quantifying the impact of the surface finish and the defect location on the detection uncertainties is also achieved.

The expanded uncertainties estimated for the depth of field and the focus position are presented for different noise standard deviations but the real standard deviation of the input image pixels is not known yet. The results presented in figures 5.3 and 5.4 indicate that if the uncertainty evaluation of the input images intensity gives a standard uncertainty higher than the limit of 7, the expected values of the focus and the depth of field might be outside the tolerance zone fixed by AV&R. Therefore, modifications of the setup might be required.

The POSIT algorithm implemented in MATLAB gives a good estimation of the camera pose considering the setup used since the average projection error is 0.60 pixels which shows a subpixel accuracy. Therefore, this method is promising if implemented onto AV&R's setup equipped with high precision tools and acquisition devices. However, the Monte Carlo simulations returned high uncertainties for rotation angles around the x and the y axis (around 23°) and a higher uncertainty for the z coordinate of the translation vector (11 mm) compared to the two other coordinates (table 5.5). By looking at the table of the sensitivity coefficients (table 5.6) one can conclude that these outliers are due to the sensitivity of the camera pose to the noise added to the 2D and 3D points. The standard deviation chosen for the control points and their images of respectively 0.5 mm and 1 pixel might be overestimated. Indeed, the only noise added in [42] is on the image points and it is a uniform noise of maximum 1 pixel. Furthermore, Zhang's method is highly sensitive to added noise as mentioned by [33]. The setup of AV&R presents less measurements uncertainties that the one used in this project. Therefore, it might be interesting to compare this pose estimation algorithm to the actual extrinsic calibration.

The calibration uncertainties have an impact on the detection algorithms hence on their uncertainties. Since AV&R's extrinsic calibration could not be entirely implemented, the associated uncertainties cannot propagate to the detection routine. Nevertheless, the inputs

of the detection routine are images acquired after the calibration. Therefore, the uncertainty of each pixel is quantified by taking a series of observations of the same zone with a random capture time between each image acquisition to take the environmental conditions into account. The first hypothesis is confirmed since the uncertainty of the pixels is high enough to notice non zero detection uncertainties. If the images had been taken without this random capture time, there would have been no or little uncertainty for the pixels. Indeed, the pixels uncertainty comes from the displacement uncertainty of the robot (repeatability), the lighting conditions, the temperature, the pressure, i.e the environmental conditions. To assess the sensitivity of the robot displacement, the same procedure could be done but without moving the robot between the acquisitions. The uncertainty proportion attributed to the robot displacement could thus be obtained.

The sensitivity of the detection outputs of *Slope Detection* was estimated for one acquisition through Spearman's correlation and feature importance. These two procedures are applicable to the rest of the acquisitions. Nevertheless, even though the default values of the three key parameters are different from one acquisition to another, the conclusions drawn should be similar. The table 5.7 shows that there is no predominant key parameter. Since our investigation did not confirm that the detection outputs are highly sensitive to those three parameters, the other parameters of *Slope Detection* need to be investigated. Their importance could thus be compared to the ones obtained for the filter size, the intensity threshold and the minimum defect size.

The preliminary version of Monte Carlo simulations implemented on the *Slope Detection* allowed to highlight the impact of the light direction and defect location on the detection uncertainties which confirms the second hypothesis. A more accurate version of Monte Carlo simulations was then implemented avoiding biased detection uncertainties. As shown in the table 5.10, the detection uncertainties are much lower for unrepaired fan blades than for repaired ones (5 times lower for the detected defects and 2.3 times lower for the false detections). The repair process might generate artifacts increasing the number of false detections but mainly disturbing the detection of defects. Nevertheless, more unrepaired fan blades are required to rerun the same procedure at a larger scale to be able to confirm those conclusions. Finally, the table 5.9 indicates that three adjacent acquisitions present higher detection uncertainties compared to the surrounding acquisitions. It does not necessary mean that there is a link between the defect location and the detection uncertainties. In fact, acquisitions 4 and 5 present few defects but the acquisition 8 has 12 defects. Since the acquisitions overlap, duplicates are not considered. Therefore, the way the duplicates are deleted might have an impact on the values of table 5.9.

6.1 Limitations and future research

Overall, the adaptation of an algorithm from one programming language to another in order to carry out Monte Carlo simulations can take a considerable amount of time which is impossible to do for an entire visual inspection system. At best, one can have an order of magnitude for each subsystem. Furthermore, the same conclusion can be drawn when implementing Monte Carlo simulations onto an industrial software not designed for this task. Also, differences in obtained results when implementing an algorithm of AV&R tends to bias the estimated uncertainty since it is not exactly the same model. It gives us however an idea of the order of magnitude of this uncertainty.

6.1.1 Calibration

As mentioned, debug images and intermediate measurements of the calibration methods designed by AV&R could have improved the results and allowed us to fully implement the Monte Carlo simulations and obtain the uncertainty level of their camera external parameters.

To quantify the pixels uncertainty of the images used in the focus calibration, one could acquire several images without the camera being calibrated similarly to section 4.3. The number of acquisitions would depend on the computation time required. The computation time of one iteration is unknown at this stage of the project. Therefore, one could compare the standard uncertainty of the input image pixels with the acceptable limit of 7. If the computed uncertainty is not within the tolerance zone, temperature and the lighting conditions in the cell need to be investigated in detail. At a constant temperature, the procedure explained in section 4.3 to compute the expected image $I_0(i, j)$ could be set up.

Even though the POSIT algorithm shows a good accuracy for the estimated pose, the computed uncertainties are too high especially for the rotation angles around the x and y axis (23°). As mentioned, our lab setup is not as accurate as the one used by AV&R. Indeed, the calibration target is machined and measured with a CMM allowing to have lower uncertainties regarding the position of the 3D control points. The same conclusion is true for the image points captured by their camera. One of the main limitations of the POSIT algorithm is that the lens distortion is not considered in the model. The camera used to implement this method has low distortion but enough to slightly deteriorate the estimated pose. However, the industrial camera used by AV&R should present an even lower distortion. Therefore, the method could be implemented and compared with the actual external calibration results once the homogeneous transformation matrix between reference frames is known. The main limitation of implementing other camera calibration techniques is the fact that using a target

with few coplanar points generally leads to a degenerate case.

6.1.2 Detection

It might be interesting to find the optimal values of the three key parameters used in the *Slope Detection* algorithm namely the filter size, the intensity threshold and the minimum defect size. For this purpose, a surrogate model needs to be built with a larger dataset free of outliers. For the moment, the significant presence of extreme values in a random generated dataset limits the analysis of the *Slope Detection* algorithm. Outliers largely present in our dataset deteriorate the fitting of a model to the data.

Two factors limit the number of Monte Carlo simulations hence the accuracy of the detection uncertainties and the compliance with standards stated by the GUM [13] : the computation time of the detection routine and the addition of the noise to the acquisitions. Reducing the cycle time could be done by removing the less sensitive steps of the detection routine or implement machine learning methods. There is obviously a minimum computation time, still consequent, required to detect and classify all the defects on numerous aeronautical parts. Instead of taking one sample of each pixel distribution per file, one could only provide the standard uncertainty of each pixel for all the *Grabs* and thus drastically reduce the required storage to run thousands of simulations. Some changes need to be made to AV&R such as the generation of several types of noise based on a given standard uncertainty. The drawback is the increase in the computation time. Moreover, to reduce the number of required simulations, Latin Hypercube Sampling could be used [88]. Moreover, it could improve the accuracy of the uncertainties since the distributions are divided into equiprobable intervals prior to the sampling contrary to Monte Carlo simulations. The parameters range is thus better taken into account.

CHAPTER 7 CONCLUSION

This thesis aimed at quantifying the uncertainty of two calibration methods and one edge detection algorithm used in the detection process of defects present near the root and leading edge of a fan blade.

The first calibration method, consisting in the identification of the focus and the depth of field of an image, is implemented in MATLAB. The uncertainty of both outputs is quantified for different added noises and since the input uncertainty of the images is not known, one determined its limit to avoid expected values being out of the tolerance zone fixed by AV&R.

Since the extrinsic camera calibration of AV&R could not be implemented due to technical and budget constraints, a pose estimation algorithm is implemented in MATLAB with a similar setup. The accuracy of the estimated pose is good but the uncertainty of the orientation is really high (around 23°). The noise added to the control points and their image is probably overestimated. Implementing this method onto AV&R's setup will drastically reduce the uncertainty (at least 6 times lower for the orientation) since the setup is really accurate compared to the one used in this work.

The pixels uncertainty of the images acquired after calibration is quantified to take into account the variability of the calibration methods into the evaluation of the detection uncertainties. The detection uncertainties of the deterministic sensitive edge detection algorithm *Slope Detection* are low (1.37 for the detected defects and 5.62 for the false detections). Regarding this algorithm, one concluded that there is no key parameter that has a predominant impact on the detection results. A dataset without outliers is required to characterize in depth the sensitivity and to determine optimal values of the three key parameters: the filter size, the threshold and the minimum defect size. The effect of defect location on the uncertainties is also presented and we determined that there is a part of the critical zone presenting higher uncertainties. Moreover, the repair of fan blades has a negative impact on the detection uncertainties since it increases them due to the addition of some artifacts.

The main drawback of the implementation of Monte Carlo simulations onto AV&R's software is the significant computation time of the detection routine. Nevertheless, the use of machine learning methods or other uncertainty propagation techniques might reduce the time needed to compute a sufficient number of simulations.

REFERENCES

[1] McKinsey & Company, "Industrial aftermarket services: Growing the core | McKinsey," [Online]. Available: https://www.mckinsey.com/industries/advanced-electronics/our-insights/industrial-aftermarket-services-growing-the-core Accessed on October 12, 2019.

[2] Generic Electric, "GE Annual Report 2018 - GE Financial Position | GE.com," [Online]. Available: https://www.ge.com/investor-relations/annual-report Accessed on October 12, 2019.

[3] Y.-H. Tsai *et al.*, "Surface defect detection of 3d objects using robot vision," *The Industrial Robot; Bedford*, vol. 38, no. 4, pp. 381–398, 2011.

[4] E. B. Njaastad, N. H. Munthe-Kaas, and O. Egeland, "Robotic Autoscanning of Highly Skewed Ship Propeller Blades," *IFAC-PapersOnLine*, vol. 51, no. 22, pp. 435–440, 2018.

[5] Z. Zhao and Y. Weng, "A flexible method combining camera calibration and hand–eye calibration," *Robotica*, vol. 31, no. 5, pp. 747–756, 2013.

[6] S. Yin *et al.*, "Development and calibration of an integrated 3d scanning system for high-accuracy large-scale metrology," *Measurement*, vol. 54, pp. 65–76, 2014.

[7] C. Yu and J. Xi, "Simultaneous and on-line calibration of a robot-based inspecting system," *Robotics and Computer-Integrated Manufacturing*, vol. 49, pp. 349 – 360, 2018.

[8] C. Gong, J. Yuan, and J. Ni, "A Self-Calibration Method for Robotic Measurement System," *Journal of Manufacturing Science and Engineering*, vol. 122, no. 1, p. 174, 2000.

[9] M. Ulrich *et al.*, "Automated and Cycle Time Optimized Path Planning for Robot-Based Inspection Systems," *Procedia CIRP*, vol. 44, pp. 377–382, 2016.

[10] B. Zitová and J. Flusser, "Image registration methods: a survey," *Image and Vision Computing*, vol. 21, no. 11, pp. 977–1000, 2003.

[11] C. M. Yeum and S. J. Dyke, "Vision-Based Automated Crack Detection for Bridge Inspection," *Computer-Aided Civil and Infrastructure Engineering*, vol. 30, no. 10, pp. 759–770, 2015.

[12] X. Li *et al.*, "Automatic registration for inspection of complex shapes," *Virtual and Physical Prototyping*, vol. 2, no. 2, pp. 75–88, 2007.

[13] Joint Committee for Guides in Metrology "JCGM 100: Evaluation of Measurement Data - Guide to the Expression of Uncertainty in Measurement," Bureau International des Poids et Mesures, 2008.

[14] "ISO 14253-2:2011(en), Geometrical product specifications (GPS) — Inspection by measurement of workpieces and measuring equipment — Part 2: Guidance for the estimation of uncertainty in GPS measurement, in calibration of measuring equipment and in product verification." [Online]. Available: https://www.iso.org/obp/ui/#iso:std:iso:14253:-2:ed-1:v1:en

[15] I. Kuselman and F. Pennecchi, "Human errors and measurement uncertainty," *Metrologia*, vol. 52, no. 2, pp. 238–243, 2015.

[16] H. Zhuang and Z. S. Roth, *Camera Aided Robot Calibration.* Boca Raton, FL, USA: CRC Press, Inc., 1995.

[17] Universal Robots. (2019), "Visor® robotic," [Online]. Available: https://www.universal-robots.com/plus/vision/visor-robotic-easy-to-use-robot-vision/ Accessed on September 24, 2019.

[18] H. Bazargani and R. Laganiere, "Camera calibration and pose estimation from planes," *IEEE Instrumentation Measurement Magazine*, vol. 18, no. 6, pp. 20–27, 2015.

[19] F. Shu, "High-precision calibration approaches to robot vision systems," thèse de doctorat, Département Informatique, Univeristé de Hambourg, Hambourg, 2009.

[20] H. Cheng and K. C. Gupta, "An Historical Note on Finite Rotations," *Journal of Applied Mechanics*, vol. 56, no. 1, pp. 139–145, 1989.

[21] J. Salvi, X. Armangué, and J. Batlle, "A comparative review of camera calibrating methods with accuracy evaluation," *Pattern Recognition*, vol. 35, no. 7, pp. 1617–1635, 2002.

[22] Guo-Qing Wei and Song De Ma, "Implicit and explicit camera calibration: theory and experiments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 5, pp. 469–480, 1994.

[23] R. Y. Tsai, "A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE J. Robotics and Automation*, vol. 3, pp. 323–344, 1987.

[24] Y. Abdel-Aziz and H. Karara, "Direct linear transformation from comparator coordinates into object space coordinates in closerange photogrammetry," *Proceedings of the Symposium on Close-Range Photogrammetry*, pp. 1–18, 1971.

[25] K. L. A. El-Ashmawy, "Using direct linear transformation (DLT) method for aerial photogrammetry applications," *Geodesy and Cartography*, vol. 44, no. 3, pp. 71–79, 2018.

[26] T. Melen, "Geometrical modelling and calibration of video cameras for underwater navigation," Ph.D. dissertation, Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, 1994.

[27] Hall *et al.*, "Measuring Curved Surfaces for Robot Vision," *Computer*, vol. 15, no. 12, pp. 42–54, 1982.

[28] J. Weng, P. Cohen, and M. Herniou, "Calibration of stereo cameras using a non-linear distortion model (CCD sensory)," in *10th International Conference on Pattern Recognition [1990] Proceedings*, vol. i, 1990, pp. 246–253 vol.1.

[29] W. Faig, "Calibration of close-range photogrammetric systems : Mathematical formulation," *Photogrammetric Engineering and Remote Sensing*, vol. 41, no. 12, 1975.

[30] I. Sobel, "On calibrating computer controlled cameras for perceiving 3-D scenes," *Artificial Intelligence*, vol. 5, no. 2, pp. 185–198, 1974.

[31] J. Salvi, "An approach to coded structured light to obtain three dimensional information," Ph.D. dissertation, Department of Electronics, Informatics and Automation, University of Girona, Girona, 1998.

[32] K. W. Wong, "Mathematical Formulation And Digital Analysis In Close-Range Photogrammetry," *Photogrammetric Engineering and Remote Sensing*, vol. 44, no. 11, 1975.

[33] W. Sun and J. R. Cooperstock, "An empirical evaluation of factors influencing camera calibration accuracy using three publicly available techniques," *Machine Vision and Applications*, vol. 17, no. 1, pp. 51–67, 2006.

[34] G. Vass and T. Perlaki, "Applying and removing lens distortion in post production," p. 8, 2003.

[35] J. Heikkila and O. Silven, "A four-step camera calibration procedure with implicit image correction," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1997, pp. 1106–1112.

[36] J. Weng, P. Cohen, and M. Herniou, "Camera calibration with distortion models and accuracy evaluation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 10, pp. 965–980, 1992.

[37] J. Heikkila, "Geometric camera calibration using circular control points," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1066–1077, 2000.

[38] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[39] R. K. Lenz and R. Y. Tsai, "Techniques for calibration of the scale factor and image center for high accuracy 3-D machine vision metrology," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 5, pp. 713–720, 1988.

[40] C. Wang, "Extrinsic calibration of a vision sensor mounted on a robot," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 2, pp. 161–175, 1992.

[41] S. Kamata *et al.*, "A camera calibration using four point-targets," in *11th IAPR International Conference on Pattern Recognition [1992] Proceedings*, 1992, pp. 550–553.

[42] D. Oberkampf, D. DeMenthon, and L. S. Davis, "Iterative Pose Estimation Using Coplanar Feature Points," *Computer Vision and Image Understanding*, vol. 63, pp. 495–511, 1996.

[43] H. Chen, "Geometry-Based Camera Calibration Using Five-Point Correspondences From a Single Image," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 12, pp. 2555–2566, 2017.

[44] C. Chatterjee and V. P. Roychowdhury, "Algorithms for coplanar camera calibration," *Machine Vision and Applications*, vol. 12, pp. 84–97, 2000.

[45] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.

[46] E. E. Hemayed, "A survey of camera self-calibration," in *[2003] Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance*, 2003, pp. 351–357.

[47] O. D. Faugeras, Q. T. Luong, and S. J. Maybank, "Camera self-calibration: Theory and experiments," in *Computer Vision — ECCV'92*, G. Sandini, Ed. Springer Berlin Heidelberg, 1992, pp. 321–334.

[48] S. J. Maybank and O. D. Faugeras, "A theory of self-calibration of a moving camera," *International Journal of Computer Vision*, vol. 8, no. 2, pp. 123–151, 1992.

[49] B. Triggs, "Autocalibration and the absolute quadric," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. San Juan, Puerto Rico: IEEE Comput. Soc, 1997, pp. 609–614.

[50] Q.-T. Luong and O. Faugeras, "Self-Calibration of a Moving Camera from Point Correspondences and Fundamental Matrices," *International Journal of Computer Vision*, vol. 22, no. 3, pp. 261–289, 1997.

[51] R. I. Hartley, "Self-calibration from multiple views with a rotating camera," in *Proceedings of the Third European Conference on Computer Vision*, ser. ECCV '94, vol. 1. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1994, pp. 471–478.

[52] G. P. Stein, "Accurate internal camera calibration using rotation, with analysis of sources of error," in *Proceedings of IEEE International Conference on Computer Vision*, 1995, pp. 230–236.

[53] T. Moons *et al.*, "Affine reconstruction from perspective image pairs with a relative object-camera translation in between," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 1, pp. 77–83, 1996.

[54] I. Miyagawa, H. Arai, and H. Koike, "Simple Camera Calibration From a Single Image Using Five Points on Two Orthogonal 1-D Objects," *IEEE Transactions on Image Processing*, vol. 19, no. 6, pp. 1528–1538, 2010.

[55] Zhengyou Zhang, "Camera calibration with one-dimensional objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 7, pp. 892–899, 2004.

[56] G. Hands and T. Armitt, *Surface and internal defect detection*. Dordrecht: Springer Netherlands, 1998, pp. 102–135.

[57] J. Aust and D. Pons, "Taxonomy of Gas Turbine Blade Defects," *Aerospace*, vol. 6, no. 5, p. 58, 2019.

[58] X. Xie, "A Review of Recent Advances in Surface Defect Detection using Texture analysis Techniques," *ELCVIA Electronic Letters on Computer Vision and Image Analysis*, vol. 7, no. 3, pp. 1–22, 2008.

[59] K. Tout, "Automatic vision system for surface inspection and monitoring: Application to wheel inspection," Ph.D. dissertation, Université de Technologie de Troyes, Troyes, 2018.

[60] H.-F. Ng, "Automatic thresholding for defect detection," *Pattern Recognition Letters*, vol. 27, no. 14, pp. 1644–1649, 2006.

[61] W. Wen and A. Xia, "Verifying edges for visual inspection purposes," *Pattern Recognition Letters*, vol. 20, no. 3, pp. 315–328, 1999.

[62] Alaknanda, R. S. Anand, and P. Kumar, "Flaw detection in radiographic weld images using morphological approach," *NDT & E International*, vol. 39, no. 1, pp. 29–33, 2006.

[63] R. C. Gonzalez and R. E. Woods, *Digital image processing*. Upper Saddle River, N.J.: Prentice Hall, 2008.

[64] M. Siegel and P. Gunatilake, "Enhanced remote visual inspection of aircraft skin," in *Proceedings of the Intelligent NDE Sciences for Aging and Futuristic Aircraft Workshop*, 1997, pp. 101 – 112.

[65] M. A. El-Agamy, M. A. M. Awad, and H. A. Sonbol, "Automated Inspection Of Surface Defects Using Machine Vision," in *Proceedings of the 17th Int. AMME Conference*, 2016.

[66] L. Wang, K. Xu, and P. Zhou, "Online Detection Technique of 3d Defects for Steel Strips Based on Photometric Stereo," in *2016 Eighth International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, 2016, pp. 428–432.

[67] R. J. Woodham, "Photometric method for determining surface orientation from multiple images," *Optical Engineering*, vol. 19, no. 1, pp. 139–144, 1980.

[68] C. Bousquet-Jette *et al.*, "Fast scene analysis using vision and artificial intelligence for object prehension by an assistive robot," *Engineering Applications of Artificial Intelligence*, vol. 63, pp. 33–44, 2017.

[69] D. Beaini *et al.*, "Computing the Spatial Probability of Inclusion inside Partial Contours for Computer Vision Applications," *arXiv:1806.01339 [cs, math]*, 2019.

[70] A. Mohebbi *et al.*, "Neural network-based decision support for conceptual design of a mechatronic system using mechatronic multi-criteria profile (MMP)," in *Proceedings of the 2014 International Conference on Innovative Design and Manufacturing (ICIDM)*, 2014, pp. 105–110.

[71] X. Tao *et al.*, "Automatic Metallic Surface Defect Detection and Recognition with Convolutional Neural Networks," *Applied Sciences*, vol. 8, no. 9, p. 1575, 2018.

[72] N. Neogi, D. K. Mohanta, and P. K. Dutta, "Review of vision-based steel surface inspection systems," *EURASIP Journal on Image and Video Processing*, vol. 2014, no. 1, p. 50, 2014.

[73] D. Weimer, B. Scholz-Reiter, and M. Shpitalni, "Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection," *CIRP Annals*, vol. 65, no. 1, pp. 417–420, 2016.

[74] J.-K. Park *et al.*, "Machine learning-based imaging system for surface defect inspection," *International Journal of Precision Engineering and Manufacturing-Green Technology*, vol. 3, no. 3, pp. 303–310, 2016.

[75] D. Beaini *et al.*, "Novel Convolution Kernels for Computer Vision and Shape Analysis based on Electromagnetism," *arXiv:1806.07996 [cs]*, 2018.

[76] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553.

[77] N. Srivastava *et al.*, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[78] B. Staar, M. Lütjen, and M. Freitag, "Anomaly detection with convolutional neural networks for industrial surface inspection," *Procedia CIRP*, vol. 79, pp. 484–489, 2019.

[79] V. Natarajan *et al.*, "Convolutional networks for voting-based anomaly classification in metal surface inspection," in *2017 IEEE International Conference on Industrial Technology (ICIT)*, 2017, pp. 986–991.

[80] Joint Committee for Guides in Metrology "JCGM 101: Evaluation of measurement data — Supplement 1 to the "Guide to the expression of uncertainty in measurement" — Propagation of distributions using a Monte Carlo method," Bureau International des Poids et Mesures, 2008.

[81] C. Coulombe *et al.*, "Applying robust design methodology to a quadrotor drone," *DS 87-4 Proceedings of the 21st International Conference on Engineering Design (ICED 17) Vol 4: Design Methods and Tools, Vancouver, Canada, 21-25.08.2017*, 2017.

[82] L. Zhu, H. Luo, and X. Zhang, "Uncertainty and sensitivity analysis for camera calibration," *Industrial Robot: An International Journal*, vol. 36, pp. 238–243, 2009.

[83] G. D. Leo and A. Paolillo, "Uncertainty evaluation of camera model parameters," in *2011 IEEE International Instrumentation and Measurement Technology Conference*, 2011, pp. 1–6.

[84] C. Ricolfe-Viala and A.-J. Sánchez-Salmerón, "Uncertainty Analysis of Camera Parameters Computed with a 3d Pattern," in *Image Analysis and Processing – ICIAP 2005*, ser. Lecture Notes in Computer Science, F. Roli and S. Vitulano, Eds. Springer Berlin Heidelberg, 2005, pp. 204–211.

[85] G. D. Leo, C. Liguori, and A. Paolillo, "Propagation of uncertainty through stereo triangulation," in *2010 IEEE Instrumentation Measurement Technology Conference Proceedings*, 2010, pp. 12–17.

[86] G. Florou and R. Mohr, "What accuracy for 3d measurements with cameras?" in *Proceedings of 13th International Conference on Pattern Recognition*, vol. 1, 1996, pp. 354–358 vol.1.

[87] H. García-Alfonso and D. Córdova-Esparza, "Comparison of uncertainty analysis of the Montecarlo and Latin Hypercube algorithms in a camera calibration model," in *2018 IEEE 2nd Colombian Conference on Robotics and Automation (CCRA)*, 2018, pp. 1–5.

[88] J. C. Helton and F. J. Davis, "Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems," *Rel. Eng. Sys. Safety*, vol. 81, pp. 23–69, 2003.

[89] M. Douilly *et al.*, "Uncertainty evaluation for pose estimation by multiple camera measurement system," in *Advanced Mathematical and Computational Tools in Metrology and Testing*, ser. Series on Advances in Mathematics for Applied Sciences. World Scientific, 2009, vol. 78, pp. 73–79.

[90] H. Zhang *et al.*, "Uncertainty Model for Template Feature Matching," in *Image and Video Technology*, ser. Lecture Notes in Computer Science. Springer International Publishing, 2018, pp. 406–420.

[91] M. D. Santo, C. Liguori, and A. Pietrosanto, "Uncertainty characterization in image-based measurements: a preliminary discussion," *IEEE Transactions on Instrumentation and Measurement*, vol. 49, no. 5, pp. 1101–1107, 2000.

[92] R. Anchini, C. Liguori, and A. Paolillo, "Evaluation of the Uncertainty of Edge-Detector Algorithms," *IEEE Transactions on Instrumentation and Measurement*, vol. 56, no. 3, pp. 681–688, 2007.

[93] C. Gillmann *et al.*, "An uncertainty-aware visual system for image pre-processing," *Journal of Imaging*, vol. 4, no. 9, 2018.

[94] G. Betta, C. Liguori, and A. Pietrosanto, "Propagation of uncertainty in a discrete Fourier transform algorithm," *Measurement*, vol. 4, no. 27, pp. 231–239, 2000.

[95] R. Krishnapuram and J. M. Keller, "Fuzzy set theoretic approach to computer vision: An overview," in *[1992 Proceedings] IEEE International Conference on Fuzzy Systems*, 1992, pp. 135–142.

[96] I. Bloch, "Fuzzy sets for image processing and understanding," *Fuzzy Sets and Systems*, vol. 281, pp. 280–291, 2015.

[97] M. Balazinski, S. Achiche, and L. Baron, "Influence of Optimization and Selection Criteria on Genetically-Generated Fuzzy Knowledge Bases," 2000.

[98] S. Achiche, M. Balazinski, and L. Baron, "Multi-combinative strategy to avoid premature convergence in genetically-generated fuzzy knowledge bases," *Journal of Theoretical and Applied Mechanics*, vol. 42, no. 3, pp. 417–444–444, 2004.

[99] I. Bloch, "Some aspects of Dempster-Shafer evidence theory for classification of multi-modality medical images taking partial volume effect into account," *Pattern Recognition Letters*, vol. 17, no. 8, pp. 905–919, 1996.

[100] S. Pertuz, M. A. Garcia, and D. Puig, "Efficient focus sampling through depth-of-field calibration," *International Journal of Computer Vision*, vol. 112, no. 3, pp. 342–353, 2015.

[101] T. Lindeberg, "Detecting salient blob-like image structures and their scales with a scale-space primal sketch: A method for focus-of-attention," *International Journal of Computer Vision*, vol. 11, pp. 283–318, 1993.

[102] Mathworks, "MATLAB - Single Calibrator App," 2019. [Online]. Available: https://www.mathworks.com/help/vision/ug/single-camera-calibrator-app.html

[103] M. Mukaka, "A guide to appropriate use of Correlation coefficient in medical research," *Malawi Medical Journal : The Journal of Medical Association of Malawi*, vol. 24, no. 3, pp. 69–71, 2012.

[104] R. Genuer, J.-M. Poggi, and C. Tuleau-Malot, "Variable selection using random forests," *Pattern Recognition Letters*, vol. 31, no. 14, pp. 2225–2236, 2010.

[105] P. Hehenberger *et al.*, "Design, modelling, simulation and integration of cyber physical systems: Methods and applications," *Computers in Industry*, vol. 82, pp. 273–289, 2016.

[106] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[107] M. D. Santo *et al.*, "Standard uncertainty evaluation in image-based measurements," *Measurement*, vol. 36, no. 3, pp. 347–358, 2004.

[108] C. Liguori, A. Paolillo, and A. Pietrosanto, "A parametric model for the uncertainty of digital images," in *XVII IMEKO World Congr.*, vol. TC2, 2003, pp. 167–172.

APPENDIX A Mathematical development of the coplanar solution for the POSIT algorithm

We consider 3 coplanar control points $\mathbf{M_0}$, $\mathbf{M_1}$ and $\mathbf{M_2}$ with $\mathbf{M_0}$ being the reference point, i.e. the origin of the *Word* reference frame. The intersection of the plane perpendicular to $\mathbf{M_0M_1}$ at $\mathbf{H_1}$ and the plane perpendicular to $\mathbf{M_0M_2}$ at $\mathbf{H_2}$ is a line. The intersection of the plane D and this line is the point $\mathbf{Q}$ defining the vector $\mathbf{I_0}$ which is the solution of the system of equations 4.17. We thus define the vector $\mathbf{w}$ normal to the plane D and along the intersection of both aforementioned planes. Geometrically, the addition of $\mathbf{I_0}$ and $\phi\mathbf{w}$ gives $\mathbf{I}$ as mentioned in equation 4.18. The same development can be done to compute $\mathbf{J}$

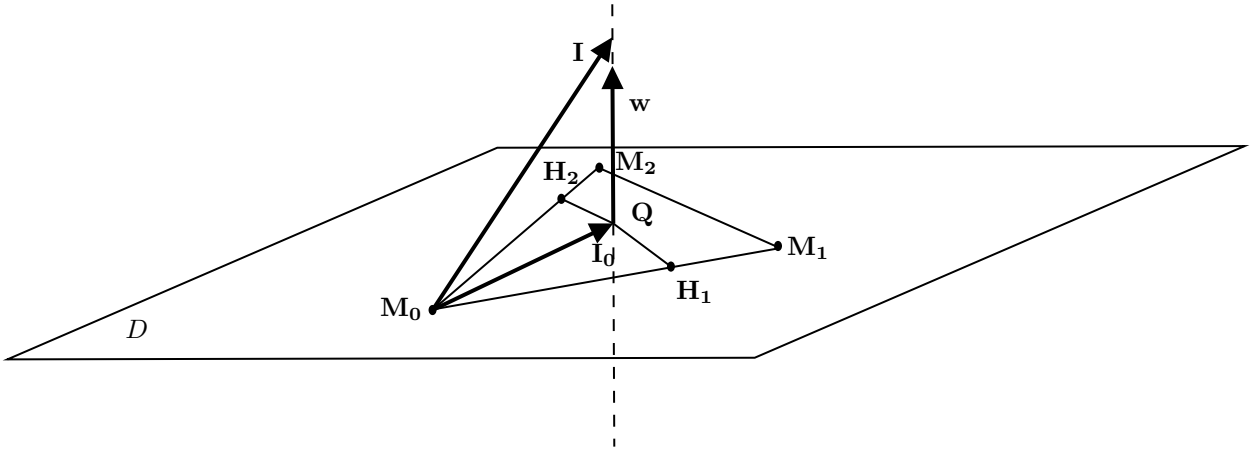


Figure A.1 Geometric interpretation of the coplanar solution reproduced with permission from [42] © 1996 Academic Press

APPENDIX B Definition of the target control points and reference frame for the POSIT algorithm

Table B.1 3D coordinates and image of the control points

Control point	2D coordinates [pixels]	3D coordinates [mm]
#1	[-149.43 -27.45]	[0 0 0]
#2	[-16.28 -26.98]	[23 0 0]
#3	[-149.03 104.63]	[0 23 0]
#4	[-17.54 106.41]	[23 23 0]

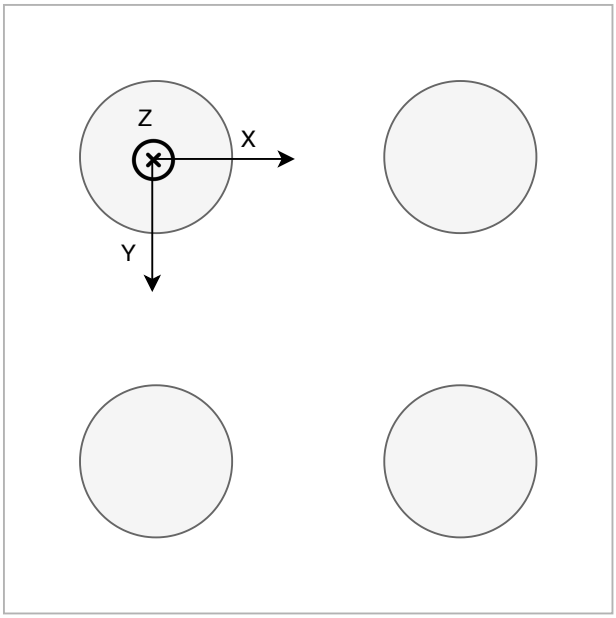


Figure B.1 Definition of the *World* reference frame